Extended Fuzzy Relations: Application to Fuzzy Expert Systems

J. J. Buckley
Mathematics Department, University of Alabama at Birmingham
Douglas Tucker
Carraway Medical Center, Birmingham, Alabama

ABSTRACT

This paper introduces the concept of an extended fuzzy relation, which is a relation whose values are vectors of fuzzy relations, some of which may also be extended fuzzy relations. Our motivation is to use extended fuzzy relations to replace blocks of rules in a fuzzy expert system with one rule. The extended fuzzy relation method is shown to contain the generalized modus ponens as a special case. The construction of extended fuzzy relations is illustrated in two examples taken from diagnosing mental disorders and image processing. We argue that the existence of an extended fuzzy relation for a block of rules may be a criterion for parallel execution of this block instead of sequential firing of the rules.

KEYWORDS: ***expert systems, fuzzy relations, pattern recognition***

1. INTRODUCTION

The purposes of this article are to introduce extended fuzzy relations (EFR), explain their application to fuzzy expert systems, and show how they may be employed to decide between parallel and sequential firing of production rules in a fuzzy expert system. We begin with defining extended fuzzy relations. Let S and T be any two nonempty sets. A regular fuzzy relation R is a function on a subset of $S \times T$ with values in [0, 1]. The strength of the relation between $s \in S$ and $t \in T$ is given by sRt, the value of R at (s, t), a number between zero and one.

An EFR $\mathcal{R}$ will be defined in a sequence of steps. The domain of $\mathcal{R}$ will be a

---

Address correspondence to J. J. Buckley, Mathematics Department, University of Alabama at Birmingham, Birmingham, Alabama 35294.

International Journal of Approximate Reasoning 1987; 1:177–195

52 Vanderbilt Ave., New York, NY 10017 0888-613X/87/$3.50

subset of $U \times V$, where $U = \{u_1, \cdots, u_m\}$ and $V = \{v_1, \cdots, v_n\}$ are two finite sets. The value of $\mathcal{R}$ at (u_i, v_j) will be denoted by $u_i \mathcal{R} v_j$.

Level Zero

We have

$$u_i \mathcal{R} v_j = \gamma_{ij} \tag{1}$$

where $\gamma_{ij} \in [0, 1]$. A level zero EFR is a regular fuzzy relation. A level zero EFR will be represented by $\mathcal{R}_0$.

Level One

For all levels ℓ, $\ell \geq 1$, the elements of U may be vectors; so let $u_i = (u_{i1}, \cdots, u_{ik_i})$, a vector of length K_i. Assume that the possible values of u_{ik} all belong to some universal set U_{ik}. Then

$$u_i \mathcal{R} v_j = R_{ij} = (R_{ij1}, \cdots, R_{ijk_i}) \tag{2}$$

where each R_{ijk} is a regular fuzzy relation $\mathcal{R}_0$. R_{ij} is a vector of length K_i of regular fuzzy relations, and the length of R_{ij} is the same as the length of u_i. A level one EFR is written as $\mathcal{R}_1$.

Level Two

We have

$$u_i \mathcal{R} v_j = R_{ij} \tag{3}$$

where R_{ij} is a vector of length K_i whose components are $\mathcal{R}_1$ or $\mathcal{R}_0$ extended fuzzy relations. A level two EFR is denoted by $\mathcal{R}_2$.

Level *L*

The value of $\mathcal{R}$ is

$$u_i \mathcal{R} v_j = R_{ij} \tag{4}$$

where the components of R_{ij} are $\mathcal{R}_\ell$, $0 \leq \ell \leq L - 1$. Let $\mathcal{R}_L$ be a level L EFR.

The definition of an EFR will be completed when we specify the domains of all the relations mentioned in the foregoing statements. The formal, recursive definition of extended fuzzy relations is rather involved, and the reader may wish to turn to an example near the end of this section which illustrates the notation.

Because U and V are finite, we may use matrix notation to describe an EFR.

Let the rows of a $m \times n$ matrix be labeled by u_i and the columns by v_j. We then place R_{ij}, or γ_{ij} for a $\mathcal{R}_0$, in the (ij)th cell. If some u_i is not related to some v_j, then we leave the corresponding (ij) cell empty. When some cells are empty, the domain of the EFR will be a proper subset of $U \times V$. Obviously, extended fuzzy relations are generalizations of regular fuzzy relations, because for levels $\ell \geq 1$ the cells contain vectors of relations.

We are primarily interested in evaluating an EFR given some data D. We will represent D as

$$D = \left\{ \frac{d_1}{u_1}, \cdots, \frac{d_m}{u_m} \right\} \tag{5}$$

where $d_i = (d_{i1}, \cdots, d_{ik_i})$ is a vector of length K_i if u_i is a vector of length K_i. Suppose each d_{ik} takes its values in some universal set $\mathfrak{D}_{ik}$. The structure of the data will vary with the levels of the extended fuzzy relations. The data for some $\mathcal{R}_\ell$ will become (generate) all the data for all the extended fuzzy relations contained in $\mathcal{R}_\ell$. Any EFR $\mathcal{R}_\ell$ under discussion will have its domain a subset of $U \times V$ with data D, but any EFR within $\mathcal{R}_\ell$ will be assumed to be defined on a subset of $\bar{U} \times \bar{V}$ with data $\bar{D}$. There may be many $\mathcal{R}_j$, $0 \leq j \leq \ell - 1$, contained in $\mathcal{R}_\ell$, but we will use the same notation $\bar{U} \times \bar{V}$ and $\bar{D}$ for all these $\mathcal{R}_j$ extended fuzzy relations.

We will now specify the domains of all the relations contained in a given EFR. For *Level Zero* $\mathcal{R}_0$ is defined on a subset of $U \times V$ with values in [0, 1]. For *Level One* each R_{ijk} is defined on a subset of $\mathfrak{D}_{ik} \times U_{ik}$ with values in [0, 1]. For *Level Two* each R_{ijk} is defined on a subset of $\mathfrak{D}_{ik} \times U_{ik}$ with values in [0, 1]. Suppose some R_{ijk} is a $\mathcal{R}_1$. This $\mathcal{R}_1$ is also defined on a subset of $\bar{U} \times \bar{V}$ with data $\bar{D}$. We require $U_{ik} \subset \bar{V}$ and $\mathfrak{D}_{ik}$ contain the possible data values $\bar{D}$ for $\mathcal{R}_1$.

Finally, for *Level L* each R_{ijk} is defined on a subset of $\mathfrak{D}_{ik} \times U_{ik}$ with values in [0, 1]. If $R_{ijk} = \mathcal{R}_\ell$, $0 \leq \ell \leq L - 1$, then $U_{ik} \subset \bar{V}$ and $\bar{D} \in \mathfrak{D}_{ik}$.

We now introduce some general notation for the evaluation of an EFR. The evaluation of $\mathcal{R}$, given D, produces output O given by

$$O = D \circ \mathcal{R} \tag{6}$$

where "$\circ$" is some type of composition of D and $\mathcal{R}$. The output O will be a fuzzy subset of V given by

$$O = \left\{ \frac{o_1}{v_1}, \cdots, \frac{o_n}{v_n} \right\} \tag{7}$$

The composition of D and $\mathcal{R}$ will be determined from the inner product $d_i * R_{ij}$ of d_i and R_{ij} and a generalized matrix product. Let

$$c_{ijk} = d_{ik} R_{ijk} u_{ik} \tag{8}$$

and

$$c_{ij} = d_i * R_{ij} = F_{ij}(c_{ij1}, \cdots, c_{ijk_i}) \tag{9}$$

where F_{ij} is some function that aggregates the c_{ijk} values into one value c_{ij}. Then we define o_j in the output O to be

$$o_j = f(c_{1j}, c_{2j}, \cdots, c_{mj}) \tag{10}$$

where f is some other function that combines the c_{ij} values into one number o_j.

The entire fuzzy output O is represented as $D \circ \mathcal{R}$, so let $D\mathcal{R}v_j = o_j$ denote a single membership value.

The evaluation procedure is easily visualized using the matrix representation for $\mathcal{R}$. Given the data $(d_i, \cdots, d_m)$, we first take the product of this vector with the column vector under v_j. The individual products of d_i and R_{ij} are defined by the inner product $d_i * R_{ij}$. We then "sum" or aggregate these inner products over an entire column into the output value o_j. Therefore, the composition $D \circ \mathcal{R}$ may be interpreted as a generalized matrix product.

Now we may present a more precise definition of an EFR.

LEVEL ZERO Let $c_{ij} = g(d_i, \gamma_{ij})$, where g is some function that combines d_i and γ_{ij} into one number c_{ij} in [0, 1]. No F_{ij} function is required, and $o_j = f(c_{ij}, \cdots, c_{mj})$.

LEVEL ONE The c_{ijk} values are obtained from $d_{ik}R_{ijk}u_{ik}$; the F_{ij} functions produce the c_{ij}, and then f gives the output o_j for all j.

LEVEL TWO When R_{ijk} is a $\mathcal{R}_0$, the c_{ijk} values are computed directly from $d_{ik}R_{ijk}u_{ik}$. So assume R_{ijk} is a $\mathcal{R}_1$. We know $U_{ik} \subset \bar{V}$, so let $u_{ik} = \bar{v}_j \in \bar{V}$. We also know d_{ik} will be a data value $\bar{D}$ for $\mathcal{R}_1$. Then

$$c_{ijk} = d_{ik}\mathcal{R}_1 u_{ik} = \bar{D}\mathcal{R}_1 \bar{v}_j = \bar{o}_j \tag{11}$$

where $\bar{o}_j$ is an output of $\mathcal{R}_1$. When R_{ijk} is a level one EFR, c_{ijk} is the correct level one output.

LEVEL L Suppose $R_{ijk} = \mathcal{R}_\ell$, $0 \leq \ell \leq L - 1$. Then

$$c_{ijk} = d_{ik}\mathcal{R}_\ell u_{ik} = \bar{D}\mathcal{R}_\ell \bar{v}_j = \bar{o}_j \tag{12}$$

where $\bar{D}$, $\bar{v}_j$, and $\bar{o}_j$ are the correct level ℓ values for $\mathcal{R}_\ell$.

Our intended use of extended fuzzy relations is to model and evaluate simultaneously blocks of rules in a fuzzy expert system. Let us consider constructing a level one EFR for the following block of rules.

$$\text{If } x_{i1}R_{ij1}A_{i1} \quad \text{and/or} \quad \cdots \; x_{ik_i}R_{ijk_i}A_{ik_i} \tag{13}$$

then y is B_j for $1 \leq i \leq m$, $1 \leq j \leq n$. Let $v_j = B_j$ and $R_{ij} = (R_{ij1}, \cdots, R_{ijk_i})$

be the vector of regular fuzzy relations. Also let $u_i = (A_{i1}, \cdots, A_{ik_i})$, $1 \le i \le m$, be a listing of the attributes on the left side. The data D contains $d_i = (x_{i1}, \cdots, x_{ik_i})$, $1 \le i \le m$. In the matrix representation of the EFR, each (ij) cell containing an R_{ij} corresponds to a rule in equation (13). Cells are left empty when a u_i and v_j are not related through a rule. The entrees in a v_j column represent all rules with the same conclusion v_j. The items in a u_i row are all the rules with the same left-side attributes. The function F_{ij} is chosen to reflect the structure (ands, ors, nots) in the left side of the rule. If only "and" is used in a rule, then we would consider using minimum for F_{ij}. We would also consider employing maximum for f because we are "oring" over all rules with the same conclusion to obtain our confidence o_j in v_j. The block of rules in equation (13) becomes, using the level one EFR,

$$\text{if } x \text{ is } D, \text{ then } y \text{ is } 0. \tag{14}$$

Example

Let us now consider an example that requires a level three EFR. This is a fuzzy expert system for diagnosing mental disorders (Siler and Tucker [8]). The information flow in this system is shown in Figure 1. The user is first asked a number of questions that, together with their answers, create a set of facts $\mathfrak{F}$. The system asks if certain behavioral manifestations exists in the patient, and the user is to enter his/her confidence, from zero to one, that they are present in the patient. A subset of these facts $\mathfrak{F}_0$ are input to a level one EFR $\mathfrak{R}_s$ that produces a fuzzy set of symptoms. Let $V_s = \{\text{depressive } (D_1), \text{ manic } (M), \text{ schizophrenic } (S_1)\}$ be the set of symptoms. The output from $\mathfrak{R}_s$ will be a fuzzy subset of V_s.

Now, $\mathfrak{R}_s$ represents a block of rules. An example of a rule from this block is as follows:

$$\begin{gathered}\text{If [fact = excess energy] AND [fact = many big plans] AND} \\ \text{[fact = hallucinations], then [symptom} = M] \end{gathered} \tag{15}$$

Let us code "excess energy" as *EE*, "many big plans" as *MBP*, and "hallucinations" as H. If the number of this rule, within this block, is seven, then $u_7 = (EE, MBP, H)$. Next assume that the user's response to questions about these conditions produced confidences $cf(EE)$, $cf(MBP)$, and $cf(H)$, respectively. The data input D_s for $\mathfrak{R}_s$ would then contain

$$\frac{(cf(EE),\ cf(MBP),\ cf(H))}{u_7} \tag{16}$$

If this rule has prior confidence 0.80 (discussed further in the section on the generalized *modus ponens*), then to evaluate $\mathfrak{R}_s$ we must first find all the c_{ij}

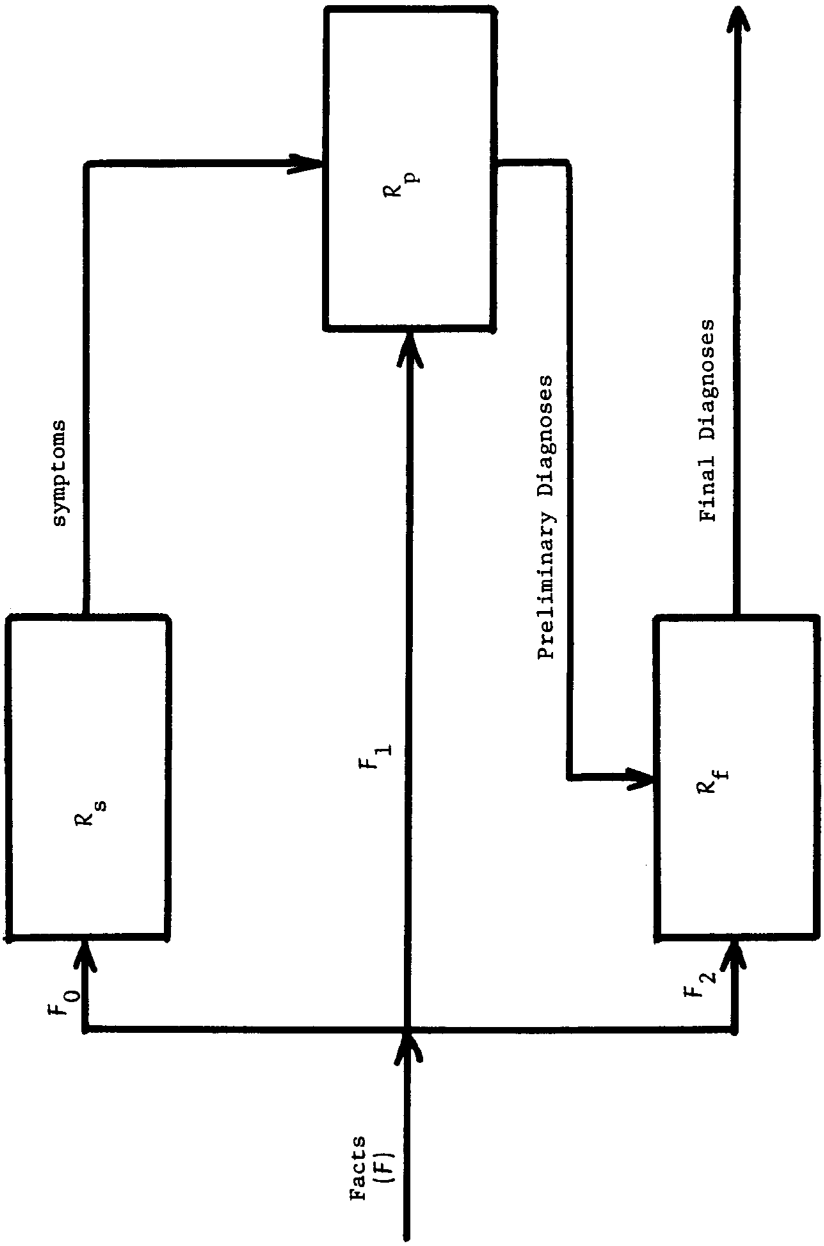

Figure 1. Information Flow in Mental Disorder Example

and, in particular,

$$c_{72} = F_{72}(cf(EE)R_{721}EE,\ cf(MBP)R_{722}MBP,\ cf(H)R_{723}H)$$
$$= \min\ (cf(EE),\ cf(MBP),\ cf(H),\ 0.80) \tag{17}$$

We have assumed the columns of $\mathcal{R}_s$ are labeled D_1, M, and S_1, so the second column is for manic. The relations in $\mathcal{R}_s$ are very simple in that they pick off the confidence in the given condition. (More complicated relations are presented in the example in the section on application.) In this way all the c_{ij} are determined, and after taking column maximums we obtain the output

$$O_s = \left\{ \frac{o_1}{D_1}, \frac{o_2}{M}, \frac{o_3}{S_1} \right\} \tag{18}$$

We next input O_s, together with more facts $\mathfrak{F}_1$, into a level two ERF $\mathcal{R}_p$ to determine a fuzzy set of preliminary diagnoses. Let V_p = {depression (D_2), schizophrenia (S_2), paranoid (P)} be the set of preliminary diagnoses. Here $\mathcal{R}_p$ also represents a block of rules, and a rule from this block is

If [symptom $= M$] AND [symptom $= D_1$] AND

[fact = persecutory or jealous delusions] AND [fact $= H$],

then [preliminary diagnosis $= D_2$] (19)

We code "persecutory or jealous delusions" as PJD. If the number of this rule is five, then $u_5 = (M, D_1, PJD, H)$, and the data D_p for $\mathcal{R}_p$ will contain

$$\frac{(\mathfrak{F}_0,\ \mathfrak{F}_0,\ cf(PJD),\ cf(H))}{u_5} \tag{20}$$

The columns in $\mathcal{R}_p$ are labeled D_2, S_2, P, so

$$c_{51} = F_{51}(\mathfrak{F}_o\mathcal{R}_sM,\ \mathfrak{F}_o\mathcal{R}_sD_1,\ cf(PJD)R_{513}PJD,\ cf(H)R_{514}H)$$
$$= \min\ (o_2,\ o_1,\ cf(PJD),\ cf(H),\ 0.90) \tag{21}$$

where 0.90 is the prior confidence in this rule. The output from $\mathcal{R}_p$ is

$$O_p = \left\{ \frac{\bar{o}_1}{D_2}, \frac{\bar{o}_2}{S_2}, \frac{\bar{o}_3}{P} \right\} \tag{22}$$

Lastly, O_p and other facts $\mathfrak{F}_2$ are put into a level three EFR $\mathcal{R}_f$ to obtain a fuzzy set of final diagnoses. Let $V_f = \{D_{31}, D_{32}, D_{33}, S_{31}, S_{32}, S_{33}, S_{34}, P_1, P_2\}$ be the collection of final diagnoses where, for example, D_{32} = manic-depressive, S_{33} = paranoid-schizophrenic, and so on. A rule from $\mathcal{R}_f$ is

If [preliminary diagnosis $= S_2$] AND [fact $= PJD$] AND

[fact $= H$], then [final diagnosis $= S_{33}$] (23)

This rule has number three, so $u_3 = (S_2, PJD, H)$ and

$$\frac{((\mathfrak{F}_o \cup \mathfrak{F}_1), cf(PJD), cf(H))}{u_3} \tag{24}$$

is in D_f. The column number for S_{33} is six, so

$$\begin{aligned} c_{36} &= F_{36}[(\mathfrak{F}_o \cup \mathfrak{F}_1)\mathcal{R}_p S_2, cf(PJD)R_{362}PJD, cf(H)R_{363}H] \\ &= \min [\bar{o}_2, cf(PJD), cf(H), 1.0] \end{aligned} \tag{25}$$

if the rule confidence is one. The final output is

$$O_f = \left\{ \frac{\bar{o}_1}{D_{31}}, \cdots, \frac{\bar{o}_9}{P_2} \right\} \tag{26}$$

Using $\mathcal{R}_f$ all the rules become

If the facts are $\mathfrak{F}$, then the (final) diagnosis is O_f (27)

Notice that multiple copies of columns from $\mathcal{R}_s$ can be in different cells in $\mathcal{R}_p$, and multiple copies of columns from $\mathcal{R}_p$ may be in many different cells in $\mathcal{R}_f$.

It may happen that we have two rules with the same attributes u_i, same conclusion $v_j = B_j$, but different relations R_{ij} and R'_{ij}. We would place both R_{ij} and R'_{ij} in the same (ij) cell but use possibly different functions F_{ij} and F'_{ij} to obtain c_{ij} and c'_{ij}, respectively, given data D. Then both c_{ij} and c'_{ij} would be input into f to obtain o_j.

The model can be easily extended to incorporate compound right sides. If rules have multiple conclusions, then we would add more columns to the EFR; each v_j would thus be a vector of conclusions. We may also consider generalizing to process multidata simultaneously. The data D were assumed to be all the information necessary for one problem (run). To process many problems at once, the data are a vector $(D_1, D_2, \cdots)$ and the output is separated into $(O_1, O_2, \cdots)$. In the example in the section of application, we process multidata simultaneously.

In the next section we compare the generalized *modus ponens* approach to the extended fuzzy relation technique for modeling fuzzy production rules. We illustrate the use of level one and level two extended fuzzy relations in an image processing problem in the section on application. We then argue, in the section of parallel versus sequential processing, that the existence of an EFR is a key to knowing when a block of rules may be executed in parallel instead of sequentially in a fuzzy expert system. The last section contains a brief summary and conclusions.

THE GENERALIZED *MODUS PONENS*

The structure of the generalized *modus ponens* (GMP) is (Zadeh [15, 16])

$$\text{if } x \text{ is } A, \text{ then } y \text{ is } B$$

$$x \text{ is } D, \text{ then } y \text{ is } O \tag{28}$$

where A and D are fuzzy subsets of $U = \{u_1, \cdots, u_m\}$ and B and O are fuzzy subsets of $V = \{v_1, \cdots, v_n\}$. One constructs, using A and B, a regular fuzzy relation R on $U \times V$ with values $u_iRv_j = \gamma_{ij}$ (Mizumoto and Zimmerman [5], Mizumoto [6], Whalen and Schott [13], Yager [14]). Then $O = D \circ R$ for some composition "$\circ$". This is exactly our level zero evaluation of an EFR.

We may also model the GMP as a level one EFR. Consider the following block of rules:

$$\text{If } x_iR_{ij}u_i, \text{ then } y \text{ is } v_j \tag{29}$$

for $1 \leq i \leq n$, $1 \leq j \leq m$. The definition of R_{ij} is

$$c_{ij} = d_iR_{ij}u_i = g(d_i, \gamma_{ij}) \tag{30}$$

and we therefore obtain the same conclusion that y is O as in the GMP.

Both the GMP and the EFR method are used to construct fuzzy sets; however, there are basic differences in the two approaches. The GMP employs a fuzzy set B, which is actually part of the rule, to obtain the fuzzy relation on $U \times V$, whereas the EFR procedure uses the structure of the left side of a block of rules to build the relation on $U \times V$. In general, an EFR cannot be modeled as a single GMP. There are three main reasons for this conclusion: (1) the EFR method allows for general input; (2) a high-level EFR is evaluated differently than a GMP; and (3) the EFR approach can incorporate prior rule confidence, thresholding, and consideration of preexisting data stored in working memory. In the GMP the d_i belong to [0, 1], as D is a fuzzy subset of U; in an EFR, however, the components of a d_i can be numbers, strings, fuzzy numbers, or discrete fuzzy sets. The only restriction on the d_i is the data types and relations allowed in the system. The example in the next section has the d_i vectors of fuzzy numbers and strings. Therefore, the EFR technique allows for general input, whereas the GMP accepts only discrete fuzzy set input. To understand the second reason given above, consider a level two EFR $\mathcal{R}_2$. Now $\mathcal{R}_2$ represents blocks of rules, and each block could possibly be modeled as a GMP. We may even be able to construct a larger GMP, say $\mathcal{P}$, to model all the blocks and $\mathcal{R}_2$. However, the evaluation of $\mathcal{R}_2$ and $\mathcal{P}$ would be different. Assume we are employing the traditional max and min for finding final confidences. Then the output from $\mathcal{R}_2$ is a fuzzy set O where o_j is the column max of a group of numbers each the min of other numbers c_{ijk}, and each c_{ijk} could be the output of a level one EFR, hence is the column max of a set of numbers each the min of other numbers. None of the suggested methods of evaluating $\mathcal{P}$ will produce the o_j numbers. The third reason will be discussed below. Modeling the GMP as a level one EFR was artificial because we never use $\gamma_{ij} = u_i R v_j$ to evaluate an EFR. That is, our fuzzy production system (Buckley, Siler, and Tucker [3]; Buckley, Siler, and Tucker [4]; Siler, Buckley, and Tucker [10], Siler, Tucker [8]) does not require a fuzzy set B to obtain O. However, we do take into consideration the preexistence of some fuzzy set for y.

Suppose, through other blocks of rules or from the initial data base, we already have in working memory y is E, where E is a fuzzy subset of V. We would not use E to evaluate the EFR, given x is D, but instead we would combine O and E into one final fuzzy subset for y. The GMP method does not allow for the existence of this fuzzy set E. We will now show how the EFR procedure can handle E together with prior rule confidence and thresholding by simply changing the F_{ij} functions.

Consider the block of rules given in equation (13) modeled by a level one EFR $\mathcal{R}_1$. Let τ_{ij}, a number between zero and one, be the prior confidence in the rule given in the (ij)th cell in $\mathcal{R}_1$. Thresholding is a procedure by which time is saved by not completely executing a rule whose left side has low confidence. Let T, a number between zero and one, be the threshold value. Suppose there already exists in working memory y is E where

$$E = \left\{ \frac{e_1}{v_1}, \cdots, \frac{e_n}{v_n} \right\} \tag{31}$$

Given that x is D, we first compute

$$c_{ijk} = d_{ik} R_{ijk} u_{ik} \tag{32}$$

and

$$\bar{c}_{ij} = d_i * R_{ij} = h_{ij}(c_{ij1}, \cdots, c_{ijk_i}) \tag{33}$$

where now h_{ij} is the function that aggregates the c_{ijk} into $\bar{c}_{ij}$. Then

$$c_{ij} = \begin{cases} e_j & \text{if } \bar{c}_{ij} < T \\ \max\,[\min\,(\bar{c}_{ij}, \tau_{ij}), e_j], & \text{if } \bar{c}_{ij} \geq T \end{cases} \tag{34}$$

The function F_{ij} is the method of obtaining the c_{ij} from the c_{ijk}, so now F_{ij} contains h_{ij} and equation (34). Given the c_{ij}, we find o_j, our final confidence in v_j, as before, using the f function.

We use the term "weakly monotonic" for the procedure given in equation (34) because it never allows a confidence to decrease. That is, we never replace a preexisting confidence e_j with a lower value. The weakly monotonic method may be summarized as follows. First, the confidence in the left side of a rule $\bar{c}_{ij}$ is tested for threshold. Second, if $\bar{c}_{ij}$ passes threshold, then the new rule confidence is the minimum of $\bar{c}_{ij}$ and the prior rule confidence τ_{ij}. Last, if the new rule confidence exceeds e_j, then the new rule confidence becomes our new confidence in v_j; otherwise, there is no change in our confidence in v_j. The final confidence in v_j is still given as o_j, a function of the c_{ij}. Of course, one may consider using other appropriate functions for max and min in equation (34).

The main difference between the GMP method and the level one EFR approach is their use of preexisting fuzzy sets for the right sides of rules. The GMP technique employs a fuzzy set, as part of the rule, for the right side of a rule in order to evaluate the rule. The EFR method, in contrast, is used to construct the fuzzy set for the right sides of a block of rules. Also, the EFR approach allows for preexisting fuzzy sets in working memory for the right sides of rules. It is our experience that the GMP is not generally applicable to rule-based fuzzy expert systems because (1) we usually do not have fuzzy sets for right sides of rules but instead wish to construct these fuzzy sets; and (2) we quite often have to deal with preexisting fuzzy sets, in working memory, for the right sides of rules.

APPLICATION

Our problem was to design an expert system to medically classify regions that were previously identified in an echocardiogram. All these details have been reported elsewhere (Buckley and Siler [2]; Siler, Tucker, Buckley, Hess and Powell [7]; Tucker, Siler, Powell, and Stanley [11]); therefore, we will be concerned here only with the facts necessary to construct fuzzy relations. Numerical feature extraction was first carried out for each region. The features of each region used are area (a fuzzy number a); x-coordinate of the centroid (a fuzzy number $\bar{x}$); y-coordinate of the centroid (a fuzzy number $\bar{y}$); type; and border. Type equals 1(0) if the pixels in the region are turned on (off), and border equals 1(0) if the region touches (does not touch) the border of the picture. Using the fuzzy numbers a, $\bar{x}$, and $\bar{y}$, we first constructed fuzzy subsets of sets SIZE, XPOS, and YPOS, respectively. We will now discuss in detail the type one EFR that goes with SIZE.

The set SIZE equals {teeny, small, medium, large, huge}. $N(r)$ denotes an appropriate fuzzy number centered at r, and LTE (GTE) are regular fuzzy relations less than or equal to (greater than of equal to) defined on fuzzy numbers. The block of rules used to build the fuzzy subset of SIZE are as follows:

If a LTE $N(100)$, then SIZE teeny

If a LTE $N(200)$ and GTE $N(100)$, then SIZE small

If a LTE $N(500)$ and GTE $N(200)$, then SIZE medium

If a LTE $N(1000)$ and GTE $N(500)$, then SIZE large

If a GTE $N(1000)$, then SIZE huge

We now define a level one EFR $\mathcal{R}_a$ to take the place of this block of rules. Let V = SIZE and $u_1 = N(100)$, $u_2 = (N(100), N(200))$, $u_3 = (N(200), N(500))$, $u_4 = (N(500), N(1000))$, and $u_5 = N(1000)$ with $U = \{u_1, \cdots, u_5\}$. The values of the R_{ij} are R_{11} = LTE, $R_{22} = R_{33} = R_{44}$ = (GTE, LTE), R_{55} = GTE, and all the (ij) cells are empty for $i \neq j$. The data D values are $d_1 = d_5 = a$ and $d_2 = d_3 = d_4 = (a, a)$. This is a very simple EFR because all the cells off the main diagonal are empty. The output $O = D \circ \mathcal{R}_a$, a fuzzy subset of V = SIZE, is given by

$$O = \left\{ \frac{o_1}{\text{teeny}}, \cdots, \frac{o_5}{\text{huge}} \right\} \tag{35}$$

where we use minimum for h_{22}, h_{33}, and h_{44}, and no f is required because $\mathcal{R}_a$ is diagonal. For example,

$$\bar{c}_{22} = \min\,(a \text{ GTE } N(100),\ a \text{ LTE } N(200)). \tag{36}$$

and

$$o_2 = \min\,(\bar{c}_{22}, \tau_{22}) \tag{37}$$

where τ_{22} is the prior confidence in this rule. The confidence o_2 will be the same confidence in "small" as obtained in the second rule in the foregoing block of rules. In constructing the fuzzy set SIZE we do not employ thresholding and the weakly monotonic procedure discussed in the preceding section. In this situation we would not have a preexisting fuzzy set for SIZE in working memory. Using $\mathcal{R}_a$, this block of rules is replaced by one rule:

$$\text{if the area is } a, \text{ then SIZE is } 0 \tag{38}$$

In a similar manner two other fuzzy subsets of XPOS = {far left, left, center, right, far right} and YPOS = {very high, high, middle, low, very low} are constructed using $\bar{x}$ and $\bar{y}$, respectively. The blocks of rules used to accomplish this may be represented by level one extended fuzzy relations $\mathcal{R}_{\bar{x}}$ and $\mathcal{R}_{\bar{y}}$, respectively.

Sometimes blocks of rules, each represented by an EFR, will form a network, with the output of some EFR being the input into another EFR. This is what occurred in the image processing problem, as shown in Figure 2. The four EFRs in Figure 2 will now be combined into one level two EFR $\mathcal{R}_2$.

The block of rules forming the primary classification may also be represented by an EFR $\mathcal{R}_c$. Instead of discussing how to define $\mathcal{R}_c$, we will concentrate on combining the EFRs into one level two EFR. A sample of the primary classification rules is as follows:

If SIZE teeny and Border = 1, then Artifact

If SIZE small and XPOS left and YPOS low, then RA

If SIZE large and YPOS low and Type = 0 and Border = 1, then Dead-Zone

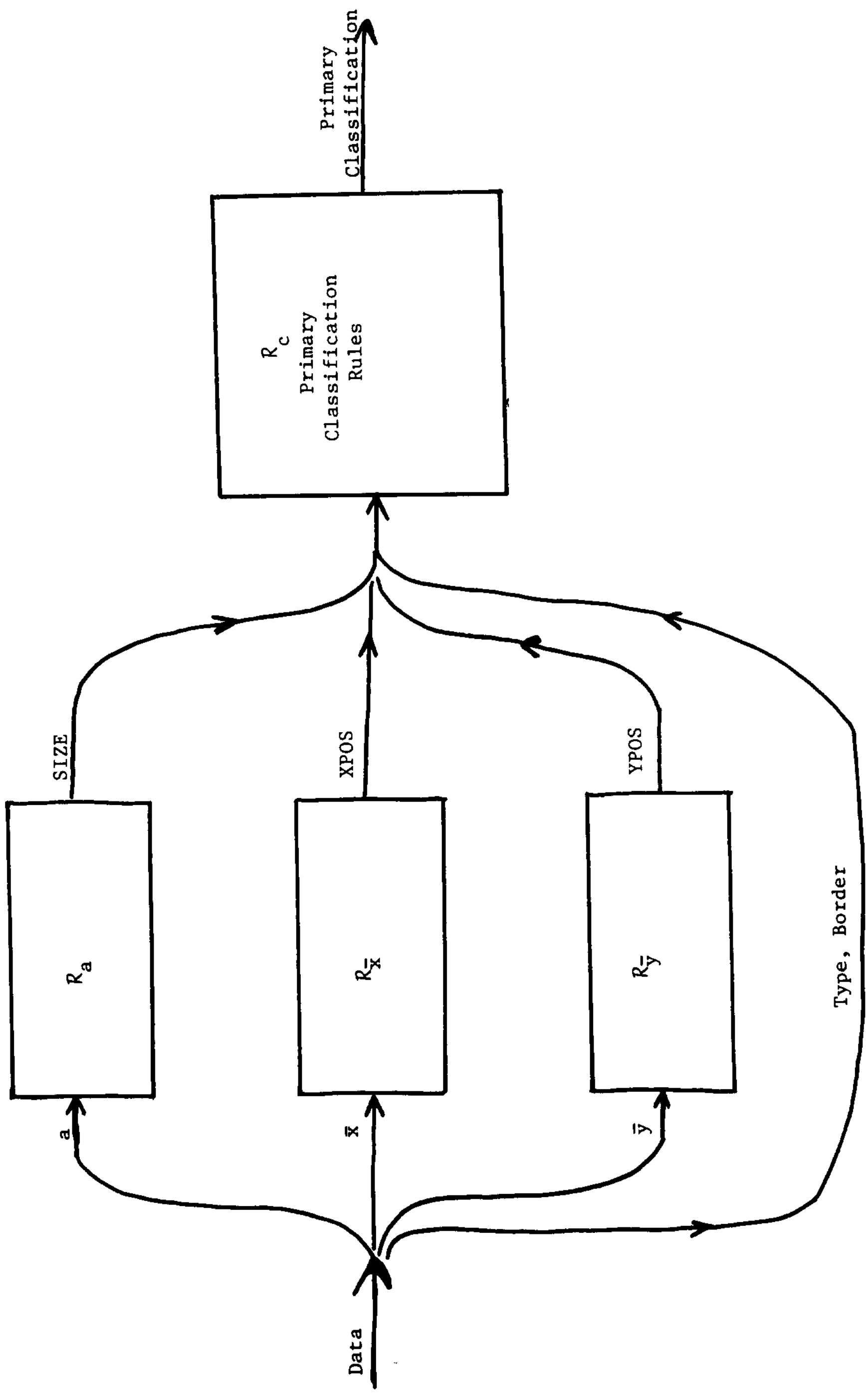

Figure 2. Network of Extended Fuzzy Relations in an Image Processing Problem

If SIZE small and XPOS right and YPOS low and Type = 0, then LA

If XPOS center and YPOS very high and Type = 0 and Border = 1, then Dead-Zone

If SIZE medium and XPOS left and YPOS high, then RV

If SIZE huge and VPOS low and Type = 0 and Border = 1, then Dead Zone

There are 11 possible classifications, which we will call $C_1, \cdots, C_{11}$ and then let $V = \{C_1, \cdots, C_{11}\}$. Each u_i is a vector, of length at most 5, whose components are members of SIZE, XPOS, or YPOS, or $u_{ik} = 1$ (0) for Type, or $u_{ik} = 1$ (0) for Border. Also, each R_{ij} is a vector whose components are $\mathcal{R}_a$, $\mathcal{R}_{\bar{x}}$, $\mathcal{R}_{\bar{y}}$, R_T, or R_B where R_T (R_B) is a binary relation for Type (Border). For the classification rules presented above, let C_1 = Artifact, $C_2 = RA$, C_3 = Dead-Zone, $C_4 = LA$, $C_5 = RV$, and

$$u_1 = (\text{teeny}, 1)$$

$$u_2 = (\text{small, left, low})$$

$$u_3 = (\text{large, low}, 0, 1)$$

$$u_4 = (\text{small, right, low}, 0)$$

$$u_5 = (\text{center, very high}, 0, 1)$$

$$u_6 = (\text{medium, left, high})$$

$$u_7 = (\text{huge, low}, 0, 1)$$

Then we see that

$$R_{11} = (\mathcal{R}_a, R_B)$$

$$R_{22} = (\mathcal{R}_a, \mathcal{R}_{\bar{x}}, \mathcal{R}_{\bar{y}})$$

$$R_{33} = (\mathcal{R}_a, \mathcal{R}_{\bar{y}}, R_T, R_B)$$

$$R_{44} = (\mathcal{R}_a, \mathcal{R}_{\bar{x}}, \mathcal{R}_{\bar{y}}, R_T)$$

$$R_{53} = (\mathcal{R}_{\bar{x}}, \mathcal{R}_{\bar{y}}, R_T, R_B)$$

$$R_{65} = (\mathcal{R}_a, \mathcal{R}_{\bar{x}}, \mathcal{R}_{\bar{y}})$$

$$R_{73} = (\mathcal{R}_a, \mathcal{R}_{\bar{y}}, R_T, R_B)$$

This is a level two EFR because the R_{ij} contain level one extended fuzzy relations.

The structure of the data elements d_i matches the structure of the u_i. For the rules presented, we see that for a given region

$$d_1 = (a, \text{Border})$$

$$d_2 = ((a, a), (\bar{x}, \bar{x}), (\bar{y}, \bar{y}))$$

$$d_3 = ((a, a), (\bar{y}, \bar{y}), \text{Type}, \text{Border})$$

$$d_4 = ((a, a), (\bar{x}, \bar{x}), (\bar{y}, \bar{y}), \text{Type})$$

$$d_5 = ((\bar{x}, \bar{x}), \bar{y}, \text{Type}, \text{Border})$$

$$d_6 = ((a, a), (\bar{x}, \bar{x}), (\bar{y}, \bar{y}))$$

$$d_7 = (a, (\bar{y}, \bar{y}), \text{Type}, \text{Border})$$

The fuzzy set of primary classifications O is determined by $D \circ \mathcal{R}_2$. We use min for the h_{ij} functions, max for f, prior rule confidence τ_{ij}, but no thresholding or weakly monotonic methods in equation (34).

Let us illustrate the method by finding o_3 for Dead-Zone. We have

$$\bar{c}_{33} = \min\ ((a, a)\mathcal{R}_a \text{ large}, (\bar{y}, \bar{y})\mathcal{R}_y \text{ low}, \text{Type } R_T 0, \text{Border } R_B 1) \quad (39)$$

and

$$c_{33} = \min\ (\bar{c}_{33}, \tau_{33}) \quad (40)$$

The value of $(a, a)\mathcal{R}_a$ large is the $\bar{o}_4$ output from the $\mathcal{R}_a$ EFR. The relation Type $R_T 0$ is one if Type $=$ 0 and zero otherwise, and Border $R_B 1$ is one if Border $=$ 1 and zero otherwise. Similarly, we compute c_{53}, c_{73}, c_{i3}, and so forth and obtain

$$o_3 = f(c_{33}, c_{53}, c_{73}, \cdots) \quad (41)$$

Using $\mathcal{R}_2$, the entire primary classification becomes one rule:

$$\text{if the data is } D\text{, then the classification is } O \quad (42)$$

In a fuzzy expert system not all rules or blocks of rules may be modeled by extended fuzzy relations. One obvious situation is where the right side of a rule calls for user interaction. Any action in the right side of a rule that cannot be interpreted as making or changing a fuzzy set will not come under the domain of extend fuzzy relations.

PARALLEL VERSUS SEQUENTIAL

An important question being asked by many computer people (both algorithm and hardware oriented) is the question of when a problem is suitable for parallel processing. In his book Uhr [12] points out that parallel processing will be an important part of future computing systems but that many obstacles must be overcome prior to widespread use of these techniques.

Through our work with a parallel rule-firing expert system (Siler, Tucker, and Buckley [9]), we have also become interested in this question. We would like to be able to quantify those characteristics which are possessed by problems suitable for parallel processing and use this as a means of discriminating between those problems which are suitable for parallel processing and those which are not.

The motivation for such work is to us obvious. It has been argued that parallel processing will result in an increase in system performance. We have shown [9] that for a particular problem, that of echocardiogram image analysis, there is an overall increase in run-time performance by a factor of 6 using a software emulation of a parallel machine.

Unfortunately, not all problems are ones that stand to benefit from use of parallel computing techniques. In a broad sense we have grouped those problems which yield to inductive reasoning as being suitable for parallel processing, and those which yield to deductive reasoning as being suitable for sequential processing. This measure is a qualitative one and gives no indication of the specific nature of the parallelism within the problem. A problem may be one that does not fit this classification scheme well, having subproblems that are both inductive and deductive in nature. What is needed is a quantitative method for expressing the "amount" of parallelism that a problem possesses and where that parallelism lies. Once this has been established, we may better apply existing techniques to solve the problem at hand.

To this end we propose that the development of an EFR or a set of related EFRs is a technique whereby the parallel nature of a problem can be demonstrated. In the preceding discussion we have shown that the evaluation of an EFR is (minimally) equivalent to the processing of several rules simultaneously. On a suitably designed machine, we think that an EFR can be effectively evaluated in one step. What we must do is to describe such a machine and show where the parallelism lies.

For such a machine, the input would be as described above, a vector of data. In a preprocessing step, copies of the data, one for each nonempty (ij) position in the EFR, would be produced and distributed to the processing elements, one processing element for each nonempty (ij) position. After this preprocessing, the evaluation of each of the inner products $d_i * R_{ij}$ and the c_{ij} would take place. In a final post-processing step, the outputs of the m processors in each column can be aggregated to produce the resulting fuzzy output O.

According to our definition of a level $\ell \geq 1$ EFR, each R_{ij} may be a vector of K_i extended fuzzy relations. For the EFR to be evaluated as efficiently as possible, each of the K_i relations R_{ijk} must be processed simultaneously. This requires that there be K_i such machines at each of the (ij) processing elements in $\mathcal{R}$. This results in a hierarchical view of the parallelism that may exist in a problem.

If we look at the echocardiogram analysis example presented earlier, we see that this is an EFR in which each of the relational components R_{ij} can contain three EFRs and two binary relations. Clearly this could be represented as a problem with two levels of parallelism, the first (lower level) being the processing of area and centroid data to generate suitable fuzzy sets for the higher-level process, the generation of the preliminary classifications (see Figure 2).

In our production system, operating in parallel mode, we emulate the machine just described. The production system's working memory acts as a central store for data that are accessible to all the R_{ijk} relations, and all those R_{ijk} relations which have suitable data are made active at once. The memory management techniques (weakly monotonic) described in the section on the generalized *modus ponens* function as the post-processing step and produce a column output for the EFR.

SUMMARY AND CONCLUSIONS

This article first introduced the idea of an extended fuzzy relation $\mathcal{R}$. The relation between two elements, say u and v, is $u\mathcal{R}v = (R_1, \cdots, R_K)$, where each R_k is a regular fuzzy relation or an extended fuzzy relation. Another type of extended fuzzy relation has been considered by Bezdek, Pettus, Stephens, and Zhang [1] and Zhang, Bezdek, Pettus, and Stephens [17], where $u\mathcal{R}v$ is a fuzzy set representing a linguistic variable. Their application is knowledge representation or information retrieval in an expert system. Our application is the use of extended fuzzy relations to replace blocks of rules in a fuzzy expert system with one rule. We have shown that our procedure contains the generalized *modus ponens* and have also shown how to construct extended fuzzy relations in two examples. We have then argued that knowing that an extended fuzzy relation exists for a block of rules may be the key to knowing when to process these rules in parallel instead of sequentially.

Firing rules in parallel, as opposed to sequentially, has two main advantages. First, a rule conflict algorithm, to determine which rule to fire when a group of rules become fireable, is not needed. Second, there is no stacking of unfired rules with subsequent backtracking. Parallel execution can result in a substantial reduction in system overhead and an increase in computational efficiency [9]. However, when operating in a parallel mode we need a memory conflict algorithm. When the system attempts to execute several rules all having the same conclusion, we need to decide on the final confidence we will place in this conclusion. Our system employs weakly monotonic logic for memory conflict resolution. The incorporation of weakly monotonic logic into the extended fuzzy relation technique has been discussed in the section on the generalized *modus ponens*.

Future research is needed to determine existence theorems for extended fuzzy relations. Results such as the following are needed for expert systems: If your problem has characteristics $\alpha, \beta, \gamma, \cdots$, then theorem $\mathfrak{I}$ says that there exists an extended fuzzy relation for this problem. Therefore, characteristics $\alpha, \beta, \gamma, \cdots$ are sufficient for putting the problem on a parallel machine.

References

1. Bezdek, J. C., Pettus, R. O., Stephens, L. M., and W.-R. Zhang, Knowledge representation using linguistic fuzzy similarity relations, *Int. J. Man-Machine Stud.* (to appear).

2. Buckley, J. J., and Siler, W., Echocardiogram analysis using fuzzy numbers and relations, *Fuzzy Sets & Syst.* (to appear).

3. Buckley, J. J., Siler, W., and Tucker, D., A fuzzy expert system, *Fuzzy Sets & Syst.* 20, 1–16, 1986.

4. Buckley, J. J., Siler, W., and Tucker, D., FLOPS, A fuzzy expert system: Applications and perspectives, in *Fuzzy Logics in Knowledge Engineering* (C. V. Negoita, and H. Prade, Eds.), Rheinland, Germany, Verlag TUV, 256–274, 1986.

5. Mizumoto, M., and Zimmermann, H.-J., Comparison of fuzzy reasoning methods, *Fuzzy Sets & Syst.* 8, 253–283, 1982.

6. Mizumoto, M., Fuzzy reasoning under new compositional rules of inference, *Kybernetics* 12, 107–117, 1985.

7. Siler, W., Tucker, D., Buckley, J. J., Hess, R. G., and Powell, V. G., Artificial intelligence in processing a sequence of time-varying images, *Proceedings of the International Society for Optical Engineering* 548, 194–199, 1985.

8. Siler, W., and Tucker, D., *FLOPS, A Fuzzy Logic Production System User's Manual*, Kemp-Carraway Heart Institute, Birmingham, Ala. 1986.

9. Siler, W., Tucker, D., and Buckley, J. J., A parallel rule firing fuzzy production system with resolution of memory conflicts by weak fuzzy monotonicity, applied to the classification of multiple objects characterized by multiple uncertain features, *Int. J. Man-Machine Studies* (to appear).

10. Siler, W., Buckley, J. J., and Tucker, D., Functional requirements for a fuzzy expert system shell, in *Artificial Intelligence: Applications of Quantitative Reasoning* (L. Zadeh and E. Sanchez, Eds.), Pergamon Press, New York, 21–31, 1987.

11. Tucker, D., Siler, W., Powell, V. G., and Stanley, A. W. H., FLOPS: A fuzzy expert system used in unsupervised enchocardiogram analysis, *Computers in Cardiology*, 341–344, 1985.

12. Uhr, L., *Algorithm-Structured Computer Arrays and Networks*, Academic Press, New York, 1984.

13. Whalen, T., and Schott, B., Alternative logics for approximate reasoning in expert systems: A comparative study, *Int. J. Man-Machine Stud.* 22, 327–346, 1985.

14. Yager, R. R., On the implication operator in fuzzy logic, *Info. Sci.* 31, 141–164, 1983.

15. Zadeh, L. A., A theory of approximate reasoning, in *Machine Intelligence 9* (J. E. Hayes, D. Michie, and L. I. Kulich, Eds.), John Wiley, New York, 149–194, 1979.

16. Zadeh, L. A., The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets & Syst.* 11, 199–227, 1983.

17. Zhang, W. R., Bezdek, J. C., Pettus, R. O., and Stephens, L. M., Linguistic fuzzy relations in expert systems, presented at the Third Annual Computer Science Symposium on Knowledge Based Systems: Theory and Applications, Univ. South Carolina, Columbia, So. Carolina, 1986.