Introducing Fuzzy Decision Tables (FDTs) for Decision Knowledge Representation and Fuzzy Decision Making

Guoqing CHEN
Department of Management Information Systems,
School of Economics & Management,
Tsinghua University, Beijing 100084, P. R. China

Abstract: This paper focuses on several aspects of the decision table methodology. Based on the (crisp) decision table (DT) formalism, fuzzy extensions are made in order to deal with imprecise and uncertain decision situations. As a result, with crisp DTs as special cases, fuzzy decision tables (FDTs) are defined, which include fuzziness in the conditions as well as in the actions. Consequently, the concept of completeness is introduced in the context of FDTs. Furthermore, fuzzy consultation of decision tables is discussed, which allows decision making with fuzziness based on the matching between fuzzy conditions and the concept of fuzzy logical implication. Finally; representing FDT knowledge in a fuzzy relational database (FRDB) environment is discussed.

1. Introduction

The motivation of fuzzy extensions to decision tables results from the fact that imprecision and uncertainty is usually involved in the process of decision making and problem solving. Originally, decision tables were used to construct the logic of programs, but more recent developments show that the application field is much larger. It is considered important that (crisp) decision tables should be formulated in a sound manner such that their usefulness is guaranteed and desirable properties are satisfied (otherwise, if a table is not complete, for example, then there will exist some possible condition combination which leads to no action or decision). In this sense, fuzzy extensions can then be made to serve the purposes of dealing with partial knowledge due to fuzziness and hence facilitating intelligent and flexible decision making.

A decision table (DT) consists of two parts: the condition part and the action part. The condition part constitutes the upper half of the table with condition subjects located on the left and condition states on the right. The action part constitutes the lower half of the table with action subjects located on the left and action subject values on the right. Condition subjects express the criteria with respect to the decision making process and action subjects describe the results of the decision making process. Formally,
Definition 1 (DT): Let CS be a condition subject with domain CD_i (i = 1, ..., cnum), CT_i be a set of condition states S_{ik} (k = 1, ..., ni , i = 1, ..., cnum) with S_{ik} being a logic expression, AS_j (j = 1, ..., anum) be an action subject; and AV_j = {true (x), false (-), nil (.)} be an action value set (j = 1, ... , anum), then a decision table (DT) is a function from $CT_1 \times CT_2 \times ... \times CT_{cnum}$ to $AV_1 \times AV_2 \times ... \times AV_{anum}$ such that each possible condition combination is mapped into one and only one action configuration.

Notably, the elements of CD_i involved in a condition state S_{ik} determine a subset of CD_i, such that the set of all these subsets constitutes a partition of CD_i.

Example 1. A crisp decision table.

1. Attitude (CS1)	Neg					Pos		
2. Performance (CS2)	<10	10-15	>15			<10	10-15	>15
3. Age (CS3)	-	-	<30	30-45	>45	-	-	-
1. Premium 1 (AS1)	x	-	x	-	x	x	-	x
2. Premium 2 (AS2)	-	x	-	x	x	x	x	x
3. Premium 3 (AS3)	-	x	x	x	-	-	x	x

In the DT, the symbol "-" appearing in the condition part denotes the irrelevance of the condition state.

DTs can be constructed according to a number of methods such as "direct method based on simple rules" or via PROLOGA (PROcedural LOGic Analyser) (Vanthienen, 1991). After the DTs are constructed, they will be checked on completeness (for each combination of conditions states there is at least one action configuration); exclusivity (for each combination of condition states there is only one action configuration); and correctness (the DT expresses what was meant by the user. If not, the decision rules need to be adapted by the designer).

As DTs can be viewed (column by column) as a set of decision rules, DTs play roles in a fast way of executing the knowledge base and in the validation and verification of the knowledge . They also show significant advantages in the knowledge acquisition phase, in which all the information has to be transformed into a coherent substance (Santos-Gomez, and Darnell, 1992; Vanthienen, and Wets, 1993).

Decision making with decision knowledge is realized by consulting DTs. Two major categories of DT consultation can be distinguished: visual consultation and transformation of DTs into a representation which is the basis for consultation in an expert system shell or program. To consult the knowledge visually, the knowledge may be retained in the DT format, such as in the case of PROLOGA. When transforming the DTs into a suitable representation for consultation in a shell or program, three options are available: integrating the decision table formalism in a relational environment (Vanthienen, and Wets, 1994), transforming DTs into decision trees (Lew, 1978), and transforming DTs into a number of rules (Vanthienen, and Wets, 1994).

2. Fuzzy Extensions to Decision Tables

Fuzzy extensions of DTs are aimed to facilitate decision making with imprecision and uncertainty which are necessary in many cases. Importantly, recent progress in formal (crisp) decision table formulation and standardization (Vanthienen, 1991; Vanthienen and Wets, 1993, 1994a, 1994b; Vanthienen and Dries, 1994) provides a sound basis on which fuzzy extensions can be made to deal with imprecise and uncertain decision situations.

2.1. Fuzzy decision tables

A crisp decision table may be extended to include fuzziness in the condition part and/or in the action part, which then gives rise to the notion of a fuzzy decision table (FDT). Fuzziness in the condition part can be expressed by fuzzy conditions (in the form of simple predicates) such as "Age is young", "Year of Service is long", etc., while fuzziness in the action part can be expressed by linguistic terms and fuzzy sets such as "Discount is high", "Add hot water", etc. In a FDT, these linguistic terms and fuzzy sets appear with condition states (S_{ik}) and/or with action subjects (AS_j). More formally, a FDT is defined as follows:

Definition 2 (FDT form 1): Let CS_i be a condition subject with domain CD_i (i = 1, ..., cnum), CT_i be a set of condition states S_{ik} (k = 1, ..., ni , i = 1, ..., cnum) with S_{ik} being a fuzzy logic expression, AS_j be an action subject incorporated with linguistic terms and fuzzy sets, and AV_j = {true (x), false (-), nil (.)} be an action value set (j = 1, ... , anum), then a fuzzy decision table (FDT) is a function from CT_1 x CT_2 x ... x CT_{cnum} to AV_1 x AV_2 x ... x AV_{anum} such that each possible condition combination is mapped into one action configuration.

Apparently, a crisp DT is a special case of a FDT. Note that in definition 2 and hereafter we assume that any fuzzy set concerned is a normalized fuzzy set, and that any condition is composed of a simple predicate of such kind as "CS_i is A" or "CS_i is in A". Moreover, when all the decision subjects involving fuzziness are of the form: "Y is B" (e.g., "Discount is small"), the FDT can be (equivalently) expressed in a form where Y is an action subject and B is one of the action subject values. In this way, a value of AS_j (j = 1, 2, ..., anum) will be not only true(x) or false(-), but also a fuzzy set or a linguistic term. Thus we have another form of FDTs:
Definition 3 (FDT form 2): Let CS_i be a condition subject with domain CD_i (i = 1, ..., cnum), CT_i be a set of condition states S_{ik} (k = 1, ..., ni , i = 1, ..., cnum) with S_{ik} being a fuzzy logic expression, AS_j be an action subject, and AV_j = {av | av is a fuzzy set of AS_j } be an action value set (j = 1, ... , anum), then a fuzzy decision table (FDT) is a function from CT_1 x CT_2 x ... x CT_{cnum} to AV_1 x AV_2 x ... x AV_{anum} such that each possible condition combination is mapped into one action configuration.

Example 2. FDTs with fuzziness in condition and action parts.

A FDT in form 1:

1. Type of Book (CS1)	hard cover				normal		
2. Wholesaler (CS2)	yes			no	-		
3. Quantity (CS3)	L	H	VH	-	L	H	VH
1. Discount small (AS1)	x	-	-	x	-	x	x
2. Discount big (AS2)	-	x	x	-	-	-	-
3. Free delivery (AS3)	-	x	x	-	-	-	x
4. Charged delivery (AS4)	-	-	-	-	x	x	-

A FDT in form 2:

1. Type of Book (CS1)	hard cover				normal		
2. Wholesaler (CS2)	yes			no	-		
3. Quantity (CS3)	L	H	VH	-	L	H	VH
1. Discount (AS1)	small	big	big	small	-	small	small
2. Delivery (AS2)	-	free	free	-	charged	charged	free

In the FDT of form 1, fuzzy sets or linguistic terms (low(L), high(H), very high(VH), small, big) appear with condition states and action subjects, while in the FDT of form 2, fuzzy sets or linguistic terms appear with condition states and action subject values. In the forthcoming discussions, for the purpose of simplicity, FDTs of these two forms will be referred to interchangeably, otherwise indicated where necessary.

The construction of FDTs can proceed mainly according to the steps of the crisp case, however, some extensions are needed. For example, extra steps are necessary to specify fuzzy sets involved in condition or actions, some provisions are needed to handle fuzzy decision rules, etc. These extensions are currently being incorporated into the PROLOGA workbench. As far as the properties of DTs (completeness, exclusivity, correctness) are concerned, it can be seen that both FDT definitions guarantee the completeness because any possible condition combination will lead to a decision in terms of action configurations. Moreover, since in general fuzzy sets appear in condition states, the degree of matching between a (given) possible condition combination and a FDT is not necessarily 0 or 1 as in the crisp case, but a value in [0,1]. Here, when talking about a condition combination with fuzziness, we refer to a fuzzy extension of the "AND" operator $\wedge_f$: $[0,1] \times [0,1] \rightarrow [0,1]$, such that $\wedge_f(0, a) = \wedge_f(a, 0) = 0$ and $\wedge_f(1, b) = \wedge_f(b, 1) = b$. Examples of such $\wedge_f$ are min (minimum) and * (multiplication). In addition, the degree of matching between two conditions can be evaluated by a closeness measure cm: $F(D) x F(D) \rightarrow [0,1]$ with cm(A, A) = 1, cm(A, B) = cm(B, A), and if supp(A) $\cap$ supp(B) = 0 then cm(A, B) = 0, where A and B are fuzzy sets in F(D) = {F | F is a fuzzy set on D}, and supp(A) and supp(B) are supports of A and B respectively.

An example of such cm is:

$$(1) \quad \sup_{x \in D} \min (A(x), B(x))$$

More formally, the concept of completeness can be defined in the context of FDTs.
Definition 4: Let FDT be a fuzzy decision table. The condition states S_{ik} (k = 1, ..., ni) of a CT_i are called complete if and only if the union of all the supports of fuzzy sets involved in all S_{ik} covers the condition domain CD_i. A FDT is called complete if and only if the condition states of each CT_i (i = 1,..., cnum) are complete.

Thus the completeness of a FDT can be guaranteed if there exists at least one column in the FDT with which the degree of matching a given condition combination is greater than zero. This is shown in the following theorem.

Theorem 1. A FDT is complete if for any given condition combination, there exists at least one column in the FDT with which the degree of matching is greater than zero.
Proof: Suppose the FDT is not complete, then according to definition 3 there exists an element e in CD_i such that e does not belong to the union of all the supports of fuzzy sets involved in S_{ik} (k = 1, ..., n_i). Now let a condition combination is merely composed of "CS_i is {1/e}", then the degree of closeness between {1/e} and any other fuzzy set F in any S_{ik} of the FDT is zero because e has a zero membership degree in F. However, this is a contradiction to the fact that the degree of matching between any condition combination and the FDT is greater than zero.

In addition, concerning the constraint for exclusivity, it has been has relaxed in FDTs. This is intuitive in many cases where multiple solutions would be possible and preferable for decision makers. For example, if some actions a1, a2, a3 and a4 should be taken merely based on the age intervals [0, 18), [18, 40), [40, 65) and [65, -) respectively, the age of "young" may lead to choosing one or more actions depending on how "young" is defined and how the matching between ages is evaluated. The notion of correctness can be determined in a similar way to that of the crisp case. That is, it can be checked by the designer whether the FDT reflects what was meant by the user.

2.2. Fuzzy decision making

Fuzzy decision making is to allow fuzzy consultation of decision tables. On one hand, fuzzy consultation can be made on crisp decision tables. This is of a great value because existing (crisp) DTs can then be utilized. On the other hand, fuzzy consultation can generally be made on fuzzy decision tables. In either case, however, a decision or action configuration cannot be taken by merely checking with each column of the table to match (perfectly) a given condition configuration. Instead, the degree of matching between the given condition combination and each column should be evaluated. As a result, more than one action configuration may be chosen, each with a degree in [0,1]. There exist various ways to derive the degree (α_l) associated with an action configuration. One way is shown in (2):

$$(2) \quad \alpha_l = \mathrm{cm}(c_1, F(S_{1k})) \wedge_f \mathrm{cm}(c_2, F(S_{2k})) \wedge_f \ldots \wedge_f \mathrm{cm}(c_{cnum}, F(S_{cnum\,k}))$$

where c = {c_1, c_2, ... , c_{cnum}} is a given condition combination with each c_i being the fuzzy set involved, $F(S_{ik})$ is a fuzzy set involved in S_{ik} appearing at column l, cm is a closeness measure, and $\wedge_f$ is a fuzzy "AND" operator as described previously. This setting says that the higher the degree of matching between condition combinations is, the higher the degree with which the corresponding action configuration is associated. This is intuitive appealing. In fact, these two degrees are set equal in (2). For example, suppose that the fuzzy set "young" for age is defined as follows:

$$(3) \qquad c_{age}(x) = \begin{cases} 1 & 0 \le x \le 17 \\ (55-x)/38 & 17 < x \le 55 \\ 0 & x > 55 \end{cases}$$

then the degree of matching between cage and the age intervals [0, 18), [18, 40), [40, 65) and [65, -) is 1, 37/38, 15/38 and 0 respectively, according to (1). Usually, a threshold λ in [0,1] may be specified when choosing actions. For instance, if $\lambda = 1$ then action a1 is chosen; if $\lambda = 0.9$ then a1 and a2 are chosen; if $\lambda = 0.35$ then a1, a2 and a3 are chosen. Notably, a lower λ means more tolerance of imprecision and uncertainty. It is worthwhile to emphasize that the above-mentioned process of fuzzy decision making is to choose those actions that are tabulated in a FDT, based upon the matching of fuzzy conditions. Further, fuzzy decision making can be dealt with under the concept of approximate reasoning, since a column of a decision table may be viewed as an if-then rule. Thus, fuzzy implication operators will play an important role. Consider the generalized modus ponens (Kerre, 1991):

if X is A then Y is B
X is A'

Y is B'

where A, B, A' and B' are fuzzy sets. In the context of a FDT, "X is A" and "Y is B" are expressed in the condition part and action part respectively, "X is A' " is a given condition, and "Y is B' " is an action to take. Note here that A' and B' are generally different from A and B. As a matter of fact, "Y is B' " is a new piece of information, knowledge or action that is derived from the FDT. In a FDT, fuzziness in the action part is modelled in terms of linguistic terms and fuzzy sets (B_j). (e.g., the person is ***heavy***), B'_j is determined using a form of so-called T-norm (e.g., $\wedge_f$) and a fuzzy implication operator I, depending on (i) the given condition combination, (ii) B_j, and (iii) the corresponding column. Concretely, let B_j and B'_j be fuzzy sets on D_j, j = 1, ..., anum, for any y in D_j,

$$(4) \quad B'_j(y) = \sup_{x \in CD1 x \ldots x CDcnum} T(A'(x), \; I(A(x), B_j(y)))$$

where I is a fuzzy implication operator, T is a T-norm (e.g., min, *, W, Z), $A'(x) = c_1(x_1) \wedge_f c_2(x_2) \wedge_f \cdots \wedge_f c_{cnum}(x_{cnum})$ is the degree of vector x corresponding to the given condition combination with each c_i being a fuzzy set involved for CS_i, and $A(x) = F(S_{1k})(x_1) \wedge_f F(S_{2k})(x_2) \wedge_f \cdots \wedge_f F(S_{cnum\ k})(x_{cnum})$ is the degree of x corresponding to the condition states of the column concerned, with each $F(S_{ik})$ being a fuzzy set involved in $F(S_{ik})$ appearing at the column. Usually, a specific I or T can be chosen considering certain intuitive knowledge. For instance, we would reasonably expect to have $B'_j = B_j$ when A' = A. This can be satisfied with T = min and I = Ig (i.e., Ig(a,b) = 1 for $a \le b$; Ig(a,b) = b for a > b). A more complete investigation on T and I can be found in Da (1991).

When consulting a FDT with fuzziness, an action configuration may be chosen with or without involving computing B'_j. If an action subject AS_j (or its value) does not involve any fuzzy set, then the action chosen is AS_j itself (or with its value), otherwise, B'_j will be derived. For example, in a FDT of form 1, if an action subject involves a fuzzy set B_j, e.g., "add hot (B_j) water", then the action configuration derived will contain an action, e.g., "add B'_j water" (with B'_j = very B_j, more-or-less B_j, etc.). The degree associated with each action configuration may also be determined using (2). Other alternatives are possible, for example, I(cm(A, A'), cm(B, B')), but need to be further explored. Finally, we will show an example with the FDT described in example 2. Suppose the given condition combination is "Type of Book is Hard cover, and the customer is a Wholesaler, and Quantity is More-or-less high". Now consider the second column. Assume:

$$A_3(q) = \text{"high"}(q) = \begin{cases} q/20 & 0 \leq q < 20 \\ 1 & \text{otherwise} \end{cases}$$

$$B_2(d) = \text{" big"}(d) = \begin{cases} d/0.2 & 0 \leq d < 0.2 \\ 1 & \text{otherwise} \end{cases}$$

Then, $A'_3(q) = \text{"more-or-less high"}(q)$

$$= \begin{cases} (q/20)1/2 & 0 \leq q < 20 \\ 1 & \text{otherwise} \end{cases}$$

Thus, using Ig for I and min for T, B'_2 can be derived according to (4):

$$B'_2(d) = \sup_q \min (A'_3(q), \quad Ig(A_3(q), B_2(d)))$$

It can be checked that B'_2 expresses something like "more-or-less big". For instance, $B'_2(0) = 0$, $B'_2(0.1) = (0.5)^{1/2} = (B_2(0.1))^{1/2}$, $B'_2(0.15) = (0.75)^{1/2} = (B_2(0.15))^{1/2}$, and $B'_2(d) = 1$ for $0.2 \leq d \leq 1$. Therefore, with this FDT, the given condition combination will lead to an action configuration "Discount is more-or-less big, and Free delivery". Using (2), the degree (α_2) associated with this action configuration is 1.

3. Representing FDTs in a Fuzzy Relational database Environment

Representing DTs in a relational database provides a mechanism to manipulate DTs using proven database techniques and functionality. The consultation of DTs can then be realized by means of traditional database queries. This section examines how the relational approach may be used to represent, store and manage FDT knowledge, which may further allow fuzzy decision making with extended SQL facilities.

3.1. Fuzzy relational databases (FRDBs)

A FRDB represents imprecise attribute values and close domain elements with possibility distributions and closeness relations respectively (Chen, Vandenbulcke, and Kerre, 1991). With the relational scheme $\mathbf{R}(A_1, A_2, \ldots, A_n)$, any n-tuple of a relation is of the form: $(\pi_{A1}, \pi_{A2}, \ldots, \pi_{An})$ where π_{Ai} is a (excluding) possibility distribution of attribute A_i on its domain D_i, and a closeness relation (reflexive and symmetric) is associated with each D_i. Based on this framework of fuzzy data representation, a number of related issues have been discussed, such as data closeness and redundancy, fuzzy functional dependency (FFD), extended relational algebra, keys and fuzzy normal forms (Chen, Kerre, and Vandenbulcke, 1992; 1993; 1994a; 1994b).

Example 3. A FRDB relation with imprecise attribute values.

Name	Sex	Age	Height	Hair-color
N1	M	25	185	black
N2	F	young	{.8/170, 1/175, 1/180/, .8/185}	{brown, red}

It is worth mentioning that the imprecision of attribute values in the tuple for N2 is reflected by a subset ({brown, red}), a linguistic term (young), and a possibility distribution ({.8/170, 1/175, 1/180/, .8/185}). In addition, closeness relations can be specified for domains (e.g., for the domain of Hair-color) to reflect the relationship between domain elements.

3.2. Managing FDT knowledge with the FRDB approach

Many efforts have been made to integrate (crisp) decision or production rules with (crisp) relational database systems in the context of decision support systems or expert database systems. In a recent study, Vanthienen and Wets (1994a) have described two such techniques. The first technique is to represent each decision table (DT) by a relational table where condition and action subjects are treated as attributes, and each decision rule is stored as a different tuple. This technique is easy to use and convenient for consultation in decision making. The second technique is to represent each DT by three relational tables for subjects, rules, and rule-parts respectively. It is based on the concept of entity-relationship (ER) methodology, and more flexible to decision situation changes. In this study, however, we will only concentrate on a fuzzy extension in accordance with the first technique.

Method (representing FDT in FRDB): When viewed vertically (column by column), a FDT can be seen as a set of ordered n-tuples of the form: $(ct_1, \ldots, ct_{cnum}, av_1, \ldots, av_{anum})$ represented in a FRDB table with the relation scheme $\mathbf{R}'(CT_1, \ldots, CT_{cnum}, AS_1, \ldots, AS_{anum})$.

Example 4. Relational tables representing the FDTs described in example 2.
The FDT in form 1 is represented in a FRDB table (R1) as follows:

Type-of-book	Whole-sale	Quantity	Discount-small	Discount-big	Free-delivery	Charged-delivery
hard-cover	yes	L	x	-	-	-
hard-cover	yes	H	-	x	x	-
hard-cover	yes	VH	-	x	x	-
hard-cover	no	-	x	-	-	-
normal	-	L	-	-	-	x
normal	-	H	x	-	-	x
normal	-	VH	x	-	x	-

The FDT in form 2 is represented in the following FRDB table (R2) where the fuzziness involved in the FDT knowledge is represented as fuzzy attribute values:

Type-of-book	Whole-sale	Quantity	Discount	Delivery
hard-cover	yes	L	small	-
hard-cover	yes	H	big	free
hard-cover	yes	VH	big	free
hard-cover	no	-	small	-
normal	-	L	-	charged
normal	-	H	small	charged
normal	-	VH	small	free

In both cases, each row of the relational table represents a column of the fuzzy decision table. Therefore the matching of fuzzy conditions in a FDT can be measured in a FRDB based on the concept of data closeness (Chen, Vandenbulcke, and Kerre, 1992). In addition, the relationship between conditions and actions in a FDT can be expressed as a cause-effect relation, to which the concept of functional dependency may apply. Readily, based upon the notion of identical functional dependency (IFD) introduced for the FRDB model in Chen, Kerre, and Vandenbulcke (1993), we will have the following IFDs:

$$(CT_1, \ldots, CT_{cnum}) \dashrightarrow_{id} AS_j \qquad j = 1, 2, \ldots, anum.$$

Furthermore, in analogue to the case of crisp databases, these IFDs can result in the notion of relation keys (hereby denoted as I-keys). Apparently, $(CT_1, \ldots, CT_{cnum})$ forms an I-key of scheme $\mathbf{R}'$. Likewise, fuzzy functional dependency (FFD) may also play a role in expressing the cause-effect relation between conditions and actions:

$$(CT_1, \ldots, CT_{cnum}) \dashrightarrow_{\theta} AS_j \qquad j = 1, 2, \ldots, anum$$

and $(CT_1, \ldots, CT_{cnum})$ forms an θ-key of $\mathbf{R}'$ (Chen, Kerre, and Vandenbulcke, 1994).

The representation of FDT knowledge in FRDB tables enables us to carry out fuzzy decision making with extended SQL facilities (e.g., SQLf (Bosc, and Pivert, 1991)). Usually, a SQL-like fuzzy query may be exemplified as follows:

```
SELECT   Discount, Delivery
FROM     R2
WHERE    Type-of-book is hard-cover AND
         Wholesaler is yes AND
         Quantity is around 30
```

For more detailed treatments for fuzzy queries, please refer to Bosc and Pivert (1991) and Bosc and Kacprzyk (1994).

4. Conclusions and Future Studies

Decision tables (DTs) are useful to represent complex decision situations in a simple fashion. Based on a sound formalism of crisp decision tables, fuzzy extensions have been made to account for the necessity of dealing with imprecision and uncertainty in decision making. First, with crisp DTs as their special cases, FDTs have been defined to allow fuzziness to be represented in both conditions and actions. The concept of completeness has then been introduced in the context of FDTs. Consequently, fuzzy decision making has been discussed, which allows consulting FDTs with linguistic terms and fuzzy sets. Some measures of fuzzy condition evaluation have been proposed, together with the treatment of action configurations derived from FDTs. Finally, in the context of fuzzy decision making and fuzzy data modelling, an approach has been proposed to represent FDTs in the FRDB environment, in that the cause-effect relations between conditions and actions of a FDT are reflected by IFDs or FFDs in fuzzy databases, and the concepts of I-keys and θ-keys apply. Thus, fuzzy decision making can be realized via fuzzy queries against FRDB tables.

Our current research and subjects of future studies include deeper explorations of closeness measures, fuzzy implication operators, T-norms, and the degrees of action configurations, in the light of desirable properties. In addition, the FDT modelling and verification in the FRDB environment with the present approach, and the exploration of corresponding issues with other representation techniques are in consideration..

References

Bosc P., and Kacprzyk J. (ed.), Studies in Fuzziness: fuzzy sets and possibility theory in database management systems. Omnitech Press, Warsaw, Springer-Verlag, Germany, 1994 (to appear).

Bosc P.; and Pivert O., About equivalents in SQLf: a relational language supporting imprecise querying, *Proceedings of International Fuzzy Engineering Symposium,* Yokohama (Japan), 309-320, 1991.

Chen G. Q., Vandenbulcke J. and Kerre E. E., A step towards the theory of fuzzy relational database design, Proc. of IFSA'91 World Congress, Brussels, 44-47, 1991.

Chen G. Q., Vandenbulcke J. and Kerre E. E., A general treatment of data redundancy in a fuzzy relational data model. Journal of The American Society for Information Science, 43, 304-311, 1992.

Chen G.Q., Kerre E.E. and Vandenbulcke J., On the lossless-join decomposition in a fuzzy relational data model. *Proceedings of International Symposium on Uncertainty Modelling & Analysis (ISUMA'93),* IEEE Press, Maryland (USA), 440-446, 1993.

Chen G.Q., Kerre E.E. and Vandenbulcke J., A computational algorithm for the FFD closure and a complete axiomatization of fuzzy functional dependency (FFD). *International Journal of Intelligent Systems,* 9, 421-439, 1994a.

Chen G.Q., Kerre E.E. and Vandenbulcke J., Fuzzy normal forms and a dependency-preserving decomposition into θ-F3NF. *Proceedings of IEEE World Congress on Computational Intelligence (WCCI: FUZZ-IEEE'94),* IEEE Press (USA), 156-161, 1994b.

Colomb R. and Chung C., Very Fast Decision Table Execution of Propositional Expert Systems, Proceedings AAAI90, 671-676, 1990.

Cragun B. and Steudel H., A Decision-Table Based processor for Checking Completeness and Consistency in Rule-Based Expert Systems, Int. Journal of Man-Machine Studies 5, 633-648, 1987.

Czogala E., Transformation of knowledge expressed by means of fuzzy decision tables into neural network representation, in Proceedings of IFSA'91 World Congress, Brussels, 48-51, 1991.

Da R., Critical study of fuzzy implication operators and their influence on approximate reasoning, in Kerre E. E. (ed.), Introduction to the basic principles of fuzzy set theory and some of its applications, Communication & Cognition, Gent, 214-251, 1991.

Dubois D. and Prade H., Fuzzy sets and systems: theory and applications, Academic Press. N.Y., 1980.

Kerre E. E. (ed.), Introduction to the basic principles of fuzzy set theory and some of its applications. Communication & Cognition, Gent, 1991 (second ed. 1994).

Lew A., Optimal Conversion of Extended-entry Decision Tables with General Cost Criteria, Communications of the ACM 4, 269-279, 1978.

Santos-Gomez L. and Darnell M., Empirical evaluation of decision tables for constructing and comprehending expert system rules, Knowledge Acquisition 4, 427-444, 1992.

Subramanian G.H., Nosek J., et al., A Comparison of the Decision Table and Tree, Communications of the ACM 1, 89-94, 1992.

Vanthienen J., Knowledge acquisition and validation using a decision table engineering workbench, The World Congress on Expert Systems, Pergamon Press, Orlando, 1861-1868, 1991.

Vanthienen J. and Dries E., Decision Tables: Refining the Concept and a Proposed Standard, Communications of the ACM, 1994 (to appear).

Vanthienen J. and Wets G., Building Intelligent Systems for Management Applications using Decision Tables, Proceedings of The Fifth Annual Conference on Intelligent Systems in Accounting, Finance and Management, Stanford, 16 pp, 1993.

Vanthienen J. and Wets G., Managing Decision Table Knowledge in a Relational Database Environment, Sixth Int. Conf. on Software Engineering and Knowledge Engineering, Latvia, 352-357, 1994a.

Vanthienen J. and Wets G., From Decision Tables to Expert System Shells, Data & Knowledge Engineering, 1994b (to appear).

Vessey I., The effect of information presentation on decision making: A cost-benefit analysis, Information & Management 27, 103-119, 1994.

Zadeh L. A., Fuzzy sets, Information and Control 8, 338-358, 1965.