Computer Science Journal of Moldova, vol.1, no.1(1), 1993

The matrix representation of fuzzy knowledge and its application to the expert systems design

V.Levchenko, A.Savinov

Abstract

An approach to the diagnostic type expert systems design based on the special matrix representation of fuzzy predicates in the attribute model of the problem domain is presented. Intensive representation of predicates by means of sectional matrices is an analogue of the conjunctive normal form. Rules, positive examples and negative examples (in general, all fuzzy) can be used to form knowledge base. Diagnostics problem is thought of as finding some attribute values provided that the information about other attribute values is available. Logical inference is based on an equivalent transformation of the matrix to that containing all prime disjuncts by using the operation $< x_k >$ of fuzzy resolution. Two strategies to carry out such transformation are described. On the basis of formalism presented the expert system shell EDIP is developed, the first version of that is non-fuzzy and the second one allows working with fuzzy data and conclusions.

1 Introduction

There exists a wide class of problem domains for which one can build (simply enough) the attribute model, i.e. to pick out a finite set of attributes essential from the point of view of the problem being solved, and to define the corresponding sets (also finite) of values for each attribute. Then the interdependencies characteristic of the diagnostic class can be described in terms of the pairs "attribute-value" and the diagnostics problem will consist in finding some attribute values if the information about other attribute values is available.

This problem converts into trivial one unless one principal limitation is taken into account: the description of the class being diagnosed for the well-known reasons is given in the intensive way, i.e. one shows the characteristic subject properties but not enumerates all subjects in the class. Besides, as we intend to deal with the human-expert knowledge, it is natural to admit the possible vagueness in the diagnostic problems (e.g. in the form of degrees of confidence) of both this very knowledge about the diagnostic class and the given initial data. The trend of our research is determined by the final goal - building in the frames of this approach the basis for the realization of the diagnostic type expert system-shell.

The knowledge is thought to be the information (possibly fuzzy) about logical ties between subject attributes in the diagnostic class. The knowledge base is presented as a number of known relationships obtained from experts' reports and other sources. The system of logical equations may be written in the special matrix form if it is represented as an analogue of the conjunctive normal form. Formalisms providing logical inference, dialog control and other conventional functions of the expert systems are based on the transformations of such matrices. The peculiarities of this matrix "packing" of predicates are employed, for the most part, for the construction of the corresponding algorithms.

One of the positive conclusions of this work is that the formalism of the fuzzy sectional vectors and the matrix representations of the fuzzy predicates make it possible to write uniformly the main information of the diagnostic problems (about the facts and rules) and to realize effectively and simply enough the logical inference algorithms, explanations and the hypotheses forming which are responsible for the functioning of an expert system.

The complete formal equality of the attribute rights in solving diagnostic problems for the system EDIP allows no selecting beforehand the goal attributes and the observed ones. The latter gives a free hand to the user during consultation. Particularly, it permits using the EDIP as the recognition or teaching system. With the help of this system-shell the manipulation with the structural (of the tree kind) knowledge base can be organized in simple ways.

2 The attribute model. The syntax and semantics of predicates

In this part we will define more exactly the notion of the attribute model of the problem domain [2] and we will give the definition of fuzzy predicates in this model, which is the base of knowledge representation. The subject (or the state) of the problem domain is compared to its description in the chosen attribute system as a set of the values of attributes. Then the consideration of the real subject is replaced by the study of its projection on this attribute system.

It is clear that the success or failure in solving a problem depends on the apt choice of the attribute system adequate to the problem. It is also obvious that the problem demanding the intricate information processing with one attribute set may be quite simple with the other set. However we will not touch upon this aspect because in most cases we do not have anything which is to be measured but something which can be measured.

All possible combinations of the values of attributes produce the description space of the problem domain. The interdependences are understood as the fact that only particular states of the problem domain are admissible. In the attribute model it means distinguishing some subset of semantically right descriptions from the whole description space.

The attribute model of the problem domain will be represented by a triple $M = < X, A, \pi >$, where X is the set of attributes, A is the collection of the sets of the values of attributes and π is the subset of the Cartesian product of the sets of values from A.

Although the attribute model definition does not show the way the set π is given in, nevertheless, as it was already said, from the point of view of application only those methods are interesting which admit the compact (intensive) representation of π. Here the elementary fuzzy predicates, defined on every from the sets of the attribute values and their logical connectives, are assumed to be the foundation of such representation. We will define exactly the elementary predicate below.

Let some problem domain be described on the syntax level by the set of n attributes $X = \{x_1, x_2, \ldots, x_n\}$, each of them taking its values from the finite sets $A_1, A_2, \ldots, A_n$ respectively. The Cartesian product of the sets of values $\Omega = A_1 \times A_2 \times \ldots \times A_n$ forms the states space of the problem domain or the universe. Every element $\omega =< e_1, e_2, \ldots, e_n >\in \Omega$ $(e_i \in A_i,\ i = 1, 2, \ldots, n)$ represents the description of some concrete state of the problem domain.

It is clear that the syntactical level of description is not enough to reflect the state of the problem domain - it also needs the semantic level which can express the problem domain relationships. To represent the semantics from the set Ω of all syntactically right descriptions, the subset of admissible descriptions should be selected. The problem of defining the subset π, as well as choosing the attribute system X, is quite nontrivial and is connected with the problem of expert knowledge elicitation which is being solved within knowledge engineering domain.

We will assume that the subset π is the fuzzy set in Zadeh's meaning [6] and we will represent it by the fuzzy predicate $\pi :\ \Omega \to [0, 1]$ in the model M. The predicate π assigns each element $\omega \in \Omega$ a number $\pi(\omega) \in [0, 1]$. It is understood, for example, as the degree of meaningfulness of the corresponding description from Ω. The predicate π imposes fuzzy restrictions on the admissible states of the problem domain, which are interpreted by the subject nature. The fuzzy predicate π may be considered to be an invariant of the problem domain remaining constant during all the time it is functioning.

The use of fuzzy knowledge in the fuzzy predicates form is especially suitable in ill structured domains, where the fuzzy semantics means can compensate the deficiency of the attribute system or the shortage of data. In any case bringing in the fuzziness always widens the expressive potential of any knowledge representation technique.

3 Sectional vectors and matrices

The sectional vector and matrix technique will be used to represent finite predicates. For the first time this technique was proposed in [1] to represent two-valued predicates, where sectional vector was defined as consisting of zeros and unities. Later [3-5,8] this method was generalized to represent fuzzy finite predicates.

Sectional vector

$$\mathbf{u} = \mathbf{u}_1.\mathbf{u}_2.\cdots.\mathbf{u}_n$$

is defined as a combination of n sections $\mathbf{u}_i$ $(i = 1, 2, \ldots, n)$ separated by points, each of them corresponding to one attribute $x_i \in X$. The section

$$\mathbf{u}_i = .\mathbf{u}_{i1}\mathbf{u}_{i2}\cdots\mathbf{u}_{in_i}.$$

consists of n_i components $\mathbf{u}_{ij}$ $(j = 1, 2, \ldots, n_i)$ taking its values from the segment $[0, 1]$. Each component corresponds to one value of the attribute. Let, for example, the problem domain be described by the attributes $X = \{COLOR,\ WEIGHT\}$ with the values from $A_1 = \{Red, Blue, Yellow, Green\}$ and $A_2 = \{Light, Heavy\}$ respectively.

Each section $\mathbf{u}_i$ of the vector $\mathbf{u}$ explicitly represents the predicate

$$\mu_i :\ A_i \to [0, 1],$$

defined over all values of the ith attribute, where

$$\mu_i(a_{ij}) = \mathbf{u}_{ij}.$$

We will refer to μ_i as elementary fuzzy predicate.

Just as in non-fuzzy logic let us define two interpretations of sectional vectors: as conjuncts and as disjuncts. The vector $\mathbf{k}$ interpreted as conjunct defines the following fuzzy predicate on Ω:

$$\mu^{\mathbf{k}}(\omega) = \min \mu^{\mathbf{k}_i}(e_i),\ where\ \omega \in \Omega,\ e_i \in A_i.$$

Similarly, if the vector $\mathbf{d}$ is interpreted as disjunct then

$$\mu^{\mathbf{d}}(\omega) = \max \mu^{\mathbf{d}_i}(e_i),\ where\ \omega \in \Omega,\ e_i \in A_i.$$

For example, the sectional vector $01\frac{1}{2}\frac{1}{7}.0\frac{1}{5}$ interpreted as conjunct defines the predicate which on the element $< Yellow, Heavy >$ equals $\min(\frac{1}{2}, \frac{1}{5}) = \frac{1}{5}$. But if this vector is interpreted as disjunct, then the predicate on the same element will be $\max(\frac{1}{2}, \frac{1}{5}) = \frac{1}{2}$.

By means of the one vector-conjunct $\mathbf{k}$ we can represent any predicate over Ω which equals 0 everywhere except the only element e, where $\mu^{\mathbf{k}}(e) = \nu > 0$. Such predicate $\mu^{\mathbf{k}}$ selects one element from the universe Ω and looks like a peak with the height ν on the zero level surface. To represent predicate $\mu^{\mathbf{k}}$ the vector $\mathbf{k}$ must contain exactly one ν in every section. With the help of this property it is possible to represent elements from Ω by the conjuncts containing one unity in every section (the rest of the components are zeros). For example, the object $< \mathit{Red}, \mathit{Light} >$ is written in the form of the conjunct 1000.10.

Similarly, by means of the one vector-disjunct $\mathbf{d}$ containing exactly one non-unity component in every section we can represent any predicate over Ω which equals 1 everywhere except the only element e, where $\mu^{\mathbf{d}}(e) = \nu < 1$ (such predicate looks like a hole in the surface of level 1.

Because of the insufficient expressive power of one vector (conjunct or disjunct) for representing any fuzzy predicates, let us introduce into consideration sectional matrices consisting of a number of sectional vector-lines. Below we will always interpret a sectional matrix $\mathbf{D}$ as a conjunction of its line-disjuncts. In accordance with this interpretation a sectional matrix $\mathbf{D}$ consisting of m lines $\mathbf{d}_i$ defines the predicate

$$\mu^{\mathbf{D}}(\omega) = \min(\mu^{\mathbf{d}_i}(\omega)) \ (i = 1, 2, \ldots, m, \ \omega \in \Omega).$$

The predicate $\mu^{\mathbf{D}}$ is an intersection of the predicates defined by the line-disjuncts of the matrix $\mathbf{D}$. The representation of fuzzy predicates in the matrix form corresponds to writing two-valued predicates in conjunctive normal form and just as in non-fuzzy case such representation is universal.

4 Equivalent transformations of sectional matrices

The minimal value of the elementary predicate defined by one section will be referred to as the section constant. Analogically, the minimal value of the predicate defined by a disjunct will be referred to as the disjunct constant:

$$\mathit{const}(\mathbf{u}_i) = \min \mu_i(a_{ij}), \ \mathit{const}(\mathbf{u}) = \min \mu^{\mathbf{u}}(\omega).$$

The section and disjunct constants will be denoted also $\mathbf{u}_{i0}$ and $\mathbf{u}_0$ respectively. The disjunct constant value is defined by its section constants in the following way:

$$const(\mathbf{u}) = \max(const(\mathbf{u}_i)).$$

If the disjunct (section) constant is equal to zero, then such disjunct (section) is said to be normal, otherwise it is said to be subnormal.

Let us introduce the following operations on the sections. Subtracting the value $\mathbf{u}$ from the section means that all its components which are less than or equal to $\mathbf{u}$ are equated to zero and the rest of components remain unchanged:

$$\check{\mathbf{u}}_{ij} = \begin{cases} \mathbf{u}_{ij} & \text{if } \mathbf{u}_{ij} > \mathbf{v} \\ 0 & \text{if } \mathbf{u}_{ij} \leq \mathbf{v} \end{cases}$$

(where $\check{\mathbf{u}}_i$ is the section with the subtracted value). Adding the value $\mathbf{v}$ to the section is defined in the contrary way. Section $\hat{\mathbf{u}}_i$ with the added to it an arbitrary value $\mathbf{v}$ is obtained by taking the componentwise maximum with this value. That is those components which are greater than $\mathbf{v}$ remain unchanged and the rest of components are equated to $\mathbf{v}$:

$$\hat{\mathbf{u}}_{ij} = \begin{cases} \mathbf{u}_{ij} & \text{if } \mathbf{u}_{ij} > \mathbf{v} \\ \mathbf{v} & \text{if } \mathbf{u}_{ij} \leq \mathbf{v}. \end{cases}$$

When adding the value $\mathbf{v}$ to the section its components can only be increased (pulled up to the level $\mathbf{v}$) and the constant will be equal $\mathbf{v}$ too. When subtracting the value $\mathbf{v}$ the section components can only be decreased (pressed down to zero) and the constant will be equal to zero.

In general, the representation of the disjunct semantics is not unique, i.e. several different disjuncts can represent the same predicate. The representation uniqueness is fulfilled only for normal disjuncts, when there is at least one element from the universe with the predicate value zero (i.e. with zero constant). But if the disjunct is subnormal, then those its components that are between zero and the constant value may be varied within this interval provided that the constant itself remains unchanged. For example, disjuncts $1011.\frac{1}{8}1$, $1\frac{1}{8}11.01$ and $1\frac{1}{8}11.\frac{1}{9}1$ are semantically equivalent, i.e. they represent the same predicate.

To remove this polysemantics let us introduce so called reduced forms of disjuncts. There exist n reduced forms for one disjunct, but for the normal disjuncts all they are the same, whereas for the subnormal ones they are different. The disjunct is said to be in kth reduced form if all its section constants except for kth one are equal to zero. Coming from this definition the way to transform any disjunct to kth reduced form is obtained. Namely, to transform disjunct $\mathbf{u}$ to kth reduced form it is necessary its constant $\mathbf{u}_0$ to add to the kth section and to subtract from the rest of the sections. This transformation results in disjunct $\check{\mathbf{u}}$, kth section of that are responsible for the saving the disjunct constant and the rest of sections are as much decreased as possible without the semantics change. For example, the disjunct $1\frac{1}{8}11.\frac{1}{9}1$ has the following two reduced forms: $1\frac{1}{8}11.01$ and $1011.\frac{1}{8}1$.

Theorem 1 $\mu^{\mathbf{u}} = \mu^{\check{\mathbf{u}}}$.

We will say that the predicate ψ is a consequence of the predicate φ ($\varphi \subseteq \psi$) iff for all elements from the universe $\omega \in \Omega$ the following is satisfied: $\varphi(\omega) \leq \psi(\omega)$. Such definition of the consequence relation coincides with the inclusion relation definition for the fuzzy sets. We will consider that the disjunct $\mathbf{v}$ is a (semantical) consequence of the disjunct $\mathbf{u}$ ($\mathbf{u} \models \mathbf{v}$) iff $\mu^{\mathbf{u}} \subseteq \mu^{\mathbf{v}}$.

It is clear that if a matrix line is a consequence of the other one of the same matrix, then it can be removed from the matrix without any change of the predicate $\mu^{\mathbf{D}}$. However, generally speaking, it is impossible to determine whether the relation $\mathbf{u} \models \mathbf{v}$ is true or false by means of only vectors $\mathbf{u}$ and $\mathbf{v}$ componentwise comparison. Nevertheless, in order to do this we may use the following theorem.

Theorem 2 *If disjuncts* $\mathbf{u}$ *and* $\mathbf{v}$ *are in* k*th reduced form and* $\forall i,j\ \mathbf{u}_{ij} \leq \mathbf{v}_{ij}$, *then* $\mathbf{u} \models \mathbf{v}$.

Let us now consider an operation $< x_k >$ of fuzzy resolution of two disjuncts on kth section. The result of this operation is new disjunct

$$\mathbf{w} = \mathbf{u} < x_k > \mathbf{v}$$

such that its kth section is equal to the conjunction of the corresponding sections from $\mathbf{u}$ and $\mathbf{v}$, and the rest of the sections are equal to the disjunction of the sections from $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{w}_{ij} = \begin{cases} \min(\mathbf{u}_{ij}, \mathbf{v}_{ij}) & \text{if } i = k \\ \max(\mathbf{u}_{ij}, \mathbf{v}_{ij}) & \text{if } i \neq k. \end{cases}$$

It should be noted that such definition of the resolution, unlike conventional one, does not impose any restrictions on the disjuncts being resolved. For the fuzzy resolution operation the property is fulfilled that the resolvent is a consequence of its parents.

Theorem 3 *If* $\mathbf{w} = \mathbf{u} < x_k > \mathbf{v}$*, then disjunct* $\mathbf{w}$ *is a consequence of two disjunct* $\mathbf{u}$ *and* $\mathbf{v}$*.*

The fuzzy resolution operation $< x_k >$ can be used to transform the matrix of disjuncts $\mathbf{D}$ by means of adding to it a new line which is the resolvent of any two disjuncts from this matrix. The theorem given above guarantees that the new disjunct will be a consequence of the matrix, and the transformation itself is equivalent.

5 The problem of finding prime disjuncts

The problem of finding prime disjuncts is rather general. It means building some other matrix equivalent to the initial one (as it was mentioned above), but which has, in some sense, logicaly independent lines - the prime disjuncts. This procedure is highly important. In particular its solution can be used to determine the degree of knowledge consistency, to organize logical inference, to solve the satisfiability problem, to seek hidden dependences in repertory grids analysis [7], to generalize knowledge [3]. The first two tasks will be considered below.

Disjunct **u** is said to be prime one for the matrix **D** iff it is a consequence of this matrix ($\mathbf{D} \models \mathbf{u}$) but is not a consequence of any other disjunct **v** which is a consequence of the matrix.

Let us consider two different ways to find the prime disjuncts for some matrix **D**. The first way is an analogue of the conventional method of finding all boolean function prime implicants. This procedure is iterative. At the beginning the resolution operation is applied to all disjunct pairs from **D** on all attributes x_k and the results are added to the matrix. Then all disjuncts which are consequences of the others are cut out from the matrix obtained in this way. After this step new matrix $\mathbf{D}_1 = R(\mathbf{D})$ is derived (R - generate and cut out procedure). At the second step the procedure R is applied to the matrix $\mathbf{D}_1$ and results in the matrix $\mathbf{D}_2 = R(\mathbf{D}_1)$ and so on. The procedure is stopped if at the current step the matrix is obtained which is equal to the previous one, i.e. all new disjuncts are cut out as the consequences of those already contained in the matrix.

The procedure described for finding prime disjuncts corresponds to the breadth-first search strategy in the space of all disjuncts, as at every step all consequences of the current matrix disjuncts are derived. Such search strategy guarantees that the shortest inference chains of the prime disjuncts will be found.

Non-fuzzy version of this algorithm is realized in the diagnostic expert system shell EDIP [2] to organize logical inference. As the experiments showed the greatest part of time in finding prime disjuncts is spent on checking whether new disjuncts are the consequences of already existing ones rather than on generating new disjuncts.

The second way to find prime disjuncts uses depth-first search strategy in the space of disjuncts. The procedure begins with the resolvating of an arbitrary pair of disjuncts from **D** and adding the result to the matrix (when adding new line, checking is performed whether it is a consequence of some other line). After that new disjunct is resolvated with some disjunct derived earlier, and so on. Thus one of two disjuncts resolvated is always the disjunct obtained at the previous step. If the search reached a deadlock, i.e. at the current step it is revealed that new disjuncts cannot be obtained, then backtracking to the previous step takes place, and an attempt is made to derive a new result. The backtracking can proceed up to the very beginning when the first pair of disjuncts was chosen. The procedure is finished when resolvating any pair of disjuncts gives the result which is then cut out from the matrix.

The procedure described finds the linear inferences [3] of prime disjuncts. Every new disjunct in such inference is derived from the last and from one of the previous disjuncts. Note that such search strategy does not guarantee finding the shortest inference chains. In addition its program implementation needs a great deal of the control data structures to maintain backtracking.

In both algorithms described the situation often occurs when the resolvent derived just now is the consequence of one of its parents i.e. one of two conditions is satisfied:

$$\mathbf{u} \models \mathbf{u} < x_k > \mathbf{v} \; \textit{or} \; \mathbf{v} \models \mathbf{u} < x_k > \mathbf{v}.$$

Resolvating such disjuncts pairs is unnecessary because the result will be then cut out from the matrix as a consequence of one of its parents.

The disjuncts $\mathbf{u}$ and $\mathbf{v}$ such that

$$\mathbf{u} \not\models \mathbf{u} < x_k > \mathbf{v} \; \textit{and} \; \mathbf{v} \not\models \mathbf{u} < x_k > \mathbf{v}$$

are said to be adjacent on the attribute x_k. Thus to cut down the prime disjuncts search tree it is necessary to resolvate only the adjacent disjuncts.

To check whether some two disjuncts are adjacent it is possible to proceed from the definition, i.e. first, to find their resolvent, and then to test whether it is a consequence of at least one of two source disjuncts. But it is also possible to take advantage of the heuristic rule which allows finding out if they are adjacent on some attribute knowing only their form (without resolvating).

Theorem 4 *If two conditions are satisfied:*
1) $\exists j_k : \; \mathbf{u}_{kj_k} > \mathbf{v}_{kj_k}$,
2) $\forall i \neq k \; \exists j_i : \; \max(\mathbf{u}_{ij_i}, \mathbf{v}_{ij_i}) < \mathbf{u}_{kj_k}$,
then the resolvent $\mathbf{w} = \mathbf{u} < x_k > \mathbf{v}$ *is not a consequence of the disjunct* $\mathbf{u}$ *(*$\mathbf{u} \not\models \mathbf{w}$*).*

To check both conditions of adjacency it is necessary to check satisfiability of the theorem conditions for every of two source disjuncts. In this case the first condition of the theorem will be reduced to the condition of fuzzy incomparability of kth sections in $\mathbf{u}$ and $\mathbf{v}$. In non-fuzzy case the condition of the fuzzy incomparability of sections corresponds to the condition of the presence of the complementary pair of literals in disjuncts resolvated (i.e. one of two disjuncts must contain literal x_k and the other its negation $\bar{x}_k$). The second theorem condition requires, that when disjuncting $n-1$ sections the resolvent constant should not be too high. In non-fuzzy case this requirement means the absence of the second complementary pair of literals in the disjuncts resolvated (otherwise the result will be valid).

The maximal value the fuzzy predicate takes over the universe will be referred to as its degree of consistency. If the predicate is represented by the matrix of disjuncts, then the degree of consistency is designated as

$$consistency(\mathbf{D}) = \max(\mu^{\mathbf{D}}(\omega)).$$

In two-valued logic the predicate may be either consistent or not. In fuzzy approach the degree of consistency takes values from the interval $[0, 1]$. The degree of consistency can be easily determined through the matrix of all prime disjuncts:

$$consistency(\mathbf{D}) = \min(\max(\mathbf{p}_{ij})).$$

If the predicate is exactly not consistent, then the matrix of all prime disjuncts will contain the only zero prime disjunct.

6 Forming a knowledge base

Let us consider, as an example, a problem domain whose objects are military aircrafts, with distinctions in purpose, altitude of flight, speed and engine type. The attribute model of this problem domain is represented by four attributes

$$X = \{PURPOSE, ALTITUDE, SPEED, ENGINE\}$$

with the values from the sets

$A_1 = \{Transport, Reconnaissance, Fighter, Bomber\},$
$A_2 = \{Low, Middle, High\},$
$A_3 = \{Low, Middle, High\},$
$A_4 = \{Screw, Jet\},$

respectively. For example, it is known that if an aircraft has a high speed then its flight altitude is certainly not low and probably not middle, but the engine is exactly of the jet type. The fighter is an exception to this rule as it can apparently have a high speed at low and middle altitudes. In addition, it can be known that transport aircrafts do not fly at high altitudes.

This short example contains three types of knowledge expressed in the natural language: the rule, the positive assertion and the negative assertion. Further a special language will be described which maintains these three basic types of knowledge. The algorithms to obtain matrix of disjuncts will be considered too.

The basic unit of this language is the elementary fuzzy proposition

$$x_i = a_{ij} : \nu_{ij},$$

which consists of the precise statement that the attribute x_i takes its particular value a_{ij} and the degree of confidence ν_{ij}, that this statement is true by means of the number from the interval $[-1, 1]$. If the degree of confidence ν_{ij} is greater than zero, then the elementary fuzzy proposition will be referred to as positive, otherwise it will be called negative. In fact the negative elementary fuzzy proposition means that the attribute can not take its value with a degree of confidence $|\nu_{ij}|$. If the degree of confidence is zero, then there is no information in the proposition, i.e. it means full uncertainty. The following statements are the examples of the elementary fuzzy propositions: $PURPOSE = Transport : 1.0$; $ALTITUDE = High : 0.7$; $SPEED = Middle : -0.5$.

Since all the values of one attribute are mutually exclusive to each other, any positive elementary fuzzy proposition

$$x_i = a_{ij} : \nu,$$

is equivalent to the set of the negative elementary propositions

$$x_i = a_{i1} : -\nu, \quad \ldots, \quad x_i = a_{ij-1} : -\nu,$$
$$x_i = a_{ij+1} : -\nu, \quad \ldots, \quad x_i = a_{in_i} : -\nu,$$

.

More complex statements of the language are constructed from the elementary fuzzy propositions and the logical connectives $\rightarrow$, $\wedge$, $\vee$ (implication, conjunction, disjunction).

The rules. The rules in problem domain will be understood as the statements consisting of the left and right parts connected with the implication symbol. The left part of the rule is the conjunction of the negative elementary propositions which are interpreted as the rule fuzzy premises. The right part is the disjunction of the positive elementary propositions which are interpreted as the rule conclusions. The whole rule is written in the form of implication

$$< -EL.PROP.1 > \wedge \ldots \wedge < -EL.PROP.n > \rightarrow$$
$$< +EL.PROP.n+1 > \vee \ldots \vee < +EL.PROP.m >,$$

where "-" means that the elementary proposition is negative and "+" - positive. Representation of the rules in the form of implications is a rather convenient and typical way to formulate general dependences in the artificial intelligence tasks.

Each implication is transformed into one line of the matrix of disjuncts D. This transformation is based on the propositional calculus identity

$$a_1 \wedge \ldots \wedge a_n \rightarrow a_{n+1} \vee \ldots \vee a_m \Leftrightarrow \bar{a}_1 \vee \ldots \vee \bar{a}_n \vee a_{n+1} \vee \ldots \vee a_m,$$

hence the fuzzy implication can be written as

$$\overline{< -EL.PROP.1 >} \vee \ldots \vee \overline{< -EL.PROP.n >} \vee$$
$$< +EL.PROP.n+1 > \vee \ldots \vee < +EL.PROP.m > .$$

Replacing negations of the negative propositions by the positive propositions we will obtain

$$< +EL.PROP.1 > \vee \ldots \vee < +EL.PROP.n > \vee$$
$$< +EL.PROP.n+1 > \vee \ldots \vee < +EL.PROP.m > .$$

Now, to transform this disjunction of the positive elementary propositions to the sectional vector, it is necessary vector components, that correspond to the elementary propositions $< +EL.PROP.i >$ $(i = 1, 2, \ldots, m)$ to equate to corresponding confidence degrees ν_i $(i = 1, 2, \ldots, m)$ assigned to these propositions and the rest of the components to equate to zero.

Let us consider turning the rule "if the aircraft has high speed, then the flight altitude is exactly not low and rather not middle" into the sectional vector. Firstly, the rule needs to be written in the form

$$SPEED = Low : -1 \wedge SPEED = Middle : -1 \;\rightarrow$$
$$ALTITUDE = Middle : 0.5 \vee ALTITUDE = High : 1.$$

Secondly, it should be represented by disjunction of the elementary propositions

$$\overline{SPEED = Low : -1} \vee \overline{SPEED = Middle : -1} \vee$$
$$ALTITUDE = Middle : 0.5 \vee ALTITUDE = High : 1,$$

and, thirdly, to replace the negation of the negative propositions by the positive ones

$$SPEED = Low : 1 \vee SPEED = Middle : 1 \vee$$
$$ALTITUDE = Middle : 0.5 \vee ALTITUDE = High : 1.$$

The vector

$$\mathbf{d} = 0000.0\frac{1}{2}1.110.00$$

is the result.

The negative assertions. The negative assertion form for knowledge representation (also called the prohibition) is used to indicate some disabled combinations of the attribute values. The addition of prohibition to the knowledge base (like adding a rule) may result in only lessening the predicate value in some point of the universe.

The prohibitions are written as implication without their right parts:

$$< -EL.PROP.1 > \wedge \ldots \wedge < -EL.PROP.m > \rightarrow .$$

The transformation of the prohibition into disjunct is made analogically to that of the rule.

For example, if "the transport aircrafts do not fly at the high altitude" is known, then this may be written as prohibition

$$\begin{aligned}&PURPOSE = Reconnaissance : -1 \wedge \\ &PURPOSE = Fighter : -1 \wedge \\ &PURPOSE = Bomber : -1 \wedge ALTITUDE = Low : -1 \wedge \\ &ALTITUDE = Middle : -1 \ \rightarrow\end{aligned}$$

and then turned into disjunct

$$\mathbf{d} = 0111.110.000.00.$$

The fewer elementary propositions are included in the prohibition, the more general it is. The most concrete prohibition form is the negative example or counter-example.

Let the matrix of disjuncts **D** contains some initial knowledge. It may be earlier formulated relationships, or certain a priori supposition about the problem domain, e.g. "everything is possible" ($\mu^{\mathbf{D}} = 1$). Including counter-example in the knowledge base results in pricking out "the hole" on the predicate surface with the depth depending on the degree of confidence.

The positive assertions. Unlike the treatment of rules and prohibitions, each of which turns into one vector-disjunct, the positive assertion originally is written in the vector-conjunct form and after that the obtained conjunct and the matrix are transformed into a new matrix. Adding a positive assertion to the knowledge base may result in only increasing the predicate values, therefore positive assertions may be used as exceptions to the rules.

We will write the positive assertions as the conjunction of the positive elementary propositions:

$$< +EL.PROP.1 > \wedge \ldots \wedge < +EL.PROP.m >$$

. To transform it into a vector-conjunct it is necessary that the components corresponding to the propositions $< +EL.PROP.i >$ $(i =$

$1, 2, \ldots, m$), be equated to the degrees of confidence ν_i ($i = 1, 2, \ldots, m$), and the rest of the components be equated to zero.

The following theorem gives us a way of transforming the pair conjunct **k** and disjunct **d** into the disjunct matrix **D** under the condition $\mu^{\mathbf{D}} = \mu^{\mathbf{d}} \cup \mu^{\mathbf{k}}$.

Theorem 5 *Disjunct* **d**, *in couple with conjunct* **k** *consisting of* n *sections, is equivalent to the matrix* **D** *which contains* n *lines* $\mathbf{d}_1, \mathbf{d}_2, \ldots,$ $\mathbf{d}_n$, *where* i*th section of the disjunct* $\mathbf{d}_i$ *is equal to the disjunction of* i*th sections* **d** *and* **k** *and the rest of the sections are equal to* $\mathbf{d}_j$ $(j \neq i)$.

With the help of the procedure given in the theorem, the conjunct **k** representing one example can be included in the matrix consisting of one disjunct **d** that will give a new matrix. In order to include the conjunct **k** in the matrix **D** consisting of a number of lines, it is necessary to perform the merging of **k** with every disjunct from **D**.

The most concrete form of the positive assertion will be called (positive) example. As a result of adding an example to the knowledge base "the peak" appears on the predicate surface. Forming knowledge base from the examples is especially suitable when knowledge is represented as a number of data. Moreover, it gives an essential advantage - the possibility of considering the degrees of confidence in its truth.

7 Organization of logical inference

Suppose we have some information about the values of the observed attributes $ALTITUDE$, $SPEED$ and $ENGINE$ and it is necessary, using the knowledge represented by the matrix **D** to define the aircraft type. To do it we should carry out the logical inference on the knowledge.

It is convenient to keep the data observed in the sectional vector **f** interpreted as conjunct which will be called the vector of facts. The vector **f** defines in the space Ω the fuzzy interval $\mu^{\mathbf{f}}$ the object description is in. When there is not any data about the state of the problem domain, the vector **f** contains only unities. If it becomes known that an attribute cannot take some value, then the corresponding component of vector $\mathbf{f}$ are decreased in proportion to the degree of confidence in this fact.

The results of the logical inference are written in the conjunct form too, which will be named the vector of conclusions and denoted as $\mathbf{c}$. The interval $\mu^{\mathbf{c}}$ defined in the space Ω by the vector $\mathbf{c}$ is always included in the interval $\mu^{\mathbf{f}}$. It is connected with the trivial proposition that any fact is a consequence of itself and therefore before carring out the logical inference the data are transferred from the vector $\mathbf{f}$ to the vector $\mathbf{c}$.

Let us formulate the problem of carring out the logical inference more strictly. Suppose there are the vector-conjunct $\mathbf{f}$ and the matrix of disjuncts $\mathbf{D}$ defining the predicates $\mu^{\mathbf{f}}$ and $\mu^{\mathbf{D}}$ respectively. It is necessary to find the minimal interval $\mu^{\mathbf{c}}$ in the space Ω as the conjunct $\mathbf{c}$, which satisfies the condition

$$\mu^{\mathbf{D}} \cap \mu^{\mathbf{f}} \subseteq \mu^{\mathbf{c}}$$

. The demand of minimality of the interval $\mu^{\mathbf{c}}$ means that if any component of the vector of conclusions is decreased then the consequence relation will not be satisfied. Note that in such sense the logical inference itself is non-fuzzy, which is connected with the definition of the consequence relation for the fuzzy predicates - either any predicate is a consequence of another, or it is not. Only the facts, knowledge and conclusions are fuzzy in the logical inference.

The demand of minimality of the interval $\mu^{\mathbf{c}}$ can be reduced to the condition of the component minimum of the vector $\mathbf{c}$, therefore the logical inference procedure can be considered as a number of elementary conclusions.

The procedure for checking whether the fuzzy statement "attribute x_i takes its jth value with the degree of confidence ν" is true will be referred to as the elementary conclusion. If this statement is true the problem domain state is restricted by the fuzzy subset $\mu^{\mathbf{g}}$, where $\mathbf{g} = \mathbf{g}_\nu(i,j)$ is the disjunct with the jth component of the ith section equal to ν ($\mathbf{g}_{ij} = \nu$) and other components being zero. In this case the following inclusion is satisfied

$$\mu^{\mathbf{D}} \cap \mu^{\mathbf{f}} \subseteq \mu^{\mathbf{g}}$$

. and hence the component $\mathbf{c}_{ij}$ of the vector $\mathbf{c}$ can be equated to ν. Thus the elementary conclusion procedure is reduced to the verification the previous relation and the logical inference - to the verification it for all possible i, j and ν.

To accomplish this verification one can transform beforehand the matrix of disjuncts $\mathbf{D}$ to that containing all prime disjuncts (the knowledge degree of consistency is revealed as well) and then the following theorem can be used.

Theorem 6 *Disjunct* $\mathbf{u}$ *is a consequence of matrix* $\mathbf{D}$ *coupled with conjunct* $\mathbf{f}$*, iff there exists such prime disjunct* $\mathbf{p}$ *of the matrix* $\mathbf{D}$ *that* $\mathbf{p} \wedge \mathbf{f} \models \mathbf{u}$.

It is clear that the reduction of logical inference to a number of the elementary conclusions is not effective because of its large quantity. Such procedure resembles the backward inference method - at the beginning all hypotheses of the kind $\mathbf{g}_\nu(i,j)$ are advanced and then they are consequently verified by means of multiple passes through the matrix $\mathbf{D}$.

Let us describe the logical inference procedure which allows finding the vector of conclusions $\mathbf{c}$ by one look-through of the matrix $\mathbf{D}$ (it must contain all prime disjuncts). This procedure is based on the assumption that the whole inference can be represented by the sequence of the inferences on all prime disjuncts from $\mathbf{D}$. First, we find all conclusions from the disjunct $\mathbf{p}_1$, then from $\mathbf{p}_2$, etc. The inference procedure is direct, i.e. when running it different hypotheses are not searched for but instantly minimally admissible values of the components are indicated.

Theorem 7 *The disjunct* $\mathbf{g} = \mathbf{g}_\nu(i,j)$ *is a consequence of the disjunct* $\mathbf{p}$ *and conjunct* $\mathbf{f}$ *iff all components of the vector* $\mathbf{p} \wedge \mathbf{f}$ *except* i*th section and also* j*th component of* i*th section are less than* ν*.*

Let us consider how the inference on one disjunct proceeds. For the sake of simplicity let us assume that the data in $\mathbf{f}$ and knowledge in $\mathbf{p}$ are consistent (vector $\mathbf{p} \wedge \mathbf{f}$ contains at least one unity). The inference begins with finding in the vector $\mathbf{p} \wedge \mathbf{f}$ the only section containing unities. If there exist more than one of such sections, then none of any conclusions can be made, and the procedure is finished (we will say that the disjunct has not worked out). If such section is found and it has the number k, then all conclusions will be written only in the kth section of the vector $\mathbf{c}$. Further it is necessary to find the maximal component among all sections of the vector $\mathbf{p} \wedge \mathbf{f}$ except kth:

$$M = \max((\mathbf{p} \wedge \mathbf{f})_{ij}),\ (i \neq k).$$

The values in the kth section of the vector $\mathbf{c}$ are changed by the rule

$$\mathbf{c}_{kj} = \begin{cases} \min(\mathbf{c}_{kj}, (\mathbf{p} \wedge \mathbf{f})_{kj}) & \text{if } (\mathbf{p} \wedge \mathbf{f})_{kj} > M \\ \min(\mathbf{c}_{kj}, M) & \text{if } (\mathbf{p} \wedge \mathbf{f})_{kj} \leq M. \end{cases}$$

8 Expert system shell EDIP

Non-fuzzy variant of the formalism described in this paper was realized in the first version of the expert system shell EDIP. The system EDIP contains two main modules: the compiler and the knowledge base interpreter. First, the compiler transforms the textual representation of the knowledge base (prepared by an expert and knowledge engineer) into the matrix form. Then the procedure of finding the prime disjuncts is started up. The compiled file of the knowledge base is formed from the character strings of the attributes and their values, the matrix of the prime disjuncts, comments to the attributes and other auxiliary information. The knowledge base interpreter (or user interface) works with the compiled knowledge base and organizes the integrated environment which provides the main operations: the data entry, the logical inference, the explanation and the advancing hypotheses. Since the time consuming process of finding the prime disjuncts is running in compilation time, there is not any delay during logical inference in consultation time.

For the data entry and the output of results unified screen form is used that consists of the attribute window and the value window. The attributes names are constantly diplayed in the attribute window. The active attribute is highlighted and all its values are displayed in the value window. The data entered by the user during the consultation are represented by the ticks near all values of attributes: if there is the tick, then the corresponding value is enabled for the active attribute, otherwise the active attribute cannot take this value (the presence of the tick corresponds to the availability of unity in the vector of facts). In the same manner the system conclusions corresponding to the current state of the vector c are displayed in the inference column.

The data are entered in the system by resetting some ticks in the value window. To obtain conclusions the function of the logical inference should be invoked, which constructs the vector of conclusions and displays it in the conclusions column of the value window.

To obtain the explanation of any fact it is necessary to refer to the explanation function, moving the cursor to the value disabled as a result of the logical inference. It finds the disjunct which has worked out earlier and reflects it in the explanation column of the attribute window in the special graphical structure form which indicates the initial data and the conclusion explained. As one fact may have several explanations, the explanation function has four subfunctions which are conditionally named: the first explanation, the next explanation, the previous explanation and the last explanation.

The hypothesis function runs similarly to the explanations with the only difference that the initial data and conclusions are suppositions but not observed facts. In other words, the hypothesis function answers the question: "What facts can lead to making a conclusion?".

In the second version of the EDIP the user's data and the conclusions in consultation regime are fuzzy. The number of fuzzy gradations is fixed and equals four. Different degrees of confidence (in the column of facts and conclusion) are reflected by different number of ticks. Three ticks mean 1, two - 0.66, one - 0.33 and the absence of ticks - 0.

9 Acknowledgments

The authors would like to thank their colleague Dr.Yu.Pechersky for the suggestions he maid and attention he paid to their work, without which the present paper could hardly be done.

References

[1] Zakrevsky A.D., The logical inference in finite predicates. Preprint No.6, The technical cybernetics Institute AS BSSR, Minsk, 1989 (Russian).

[2] Levchenko V.I., The diagnostic system based on the finite predicates. Preprint, Institute of mathematics and CC AS Moldova, Kishinev, 1990 (Russian).

[3] Savinov A.A., The matrix representation of fuzzy knowledge in the attribute models. Preprint, Institute of mathematics and CC AS Moldova, Kishinev, 1991 (Russian).

[4] Levchenko V.I., Savinov A.A., The representation of fuzzy predicates and logical inference in the attribute models. Fuzzy Systems and AI Magazine, Vol.1, No.3-4, 12-15, 1991.

[5] Levchenko V.I., Savinov A.A., The matrix representation of knowledge and the solution of the diagnosis problem in the fuzzy attribute model. Fuzzy Systems and AI Magazine, Vol.1, No.3-4, 3-5, 1991.

[6] Kaufmann A., Introduction a la theorie des sous-ensembles flous (Fuzzy set theory), Masson, Paris-New York-Barcelone-Milan, 1977.

[7] Savinov A.A., Fuzzy finite predicates and the repertory grids method. In "The knowledge based application systems", Kishinev, "Shtiintsa", 1992 (Russian).

[8] Levchenko V.I., Savinov A.A., The matrix representation of the fuzzy predicates in the attribute models and its application in the

diagnostic expert systems. In "The knowledge based application systems", Kishinev, "Shtiintsa", 1992 (Russian).

[9] Chang C.L., Lee R.C.T., Symbolik logic and mechanical theorem proving, Academic press, 1973.

V.Levchenko, A.Savinov
Institute of Mathematics,
Academy of Sciences of Moldova,
5 Academiei str.,
Chişinău, 277028, Moldova
Phone: (373+2) 73-81-30
E-mail: 23lsii@math.moldova.su

Received September 15, 1992