Multivalued Knowledge-Base based on multivalued Datalog

Agnes Achs

Abstract—The basic aim of our study is to give a possible model for handling uncertain information. This model is worked out in the framework of DATALOG. The concept of multivalued knowledge-base will be defined as a quadruple of any background knowledge; a deduction mechanism; a connecting algorithm, and a function set of the program, which help us to determine the uncertainty levels of the results. At first the concept of fuzzy Datalog will be summarized, then its extensions for intuitionistic- and interval-valued fuzzy logic is given and the concept of bipolar fuzzy Datalog is introduced. Based on these extensions the concept of multivalued knowledge-base will be defined. This knowledge-base can be a possible background of a future agent-model.

Keywords—Fuzzy-, intuitionistic-, bipolar Datalog, multivalued knowledge-base

I. Introduction

A large part of human knowledge can not be modeled by pure inference systems, because this knowledge is often ambiguous, incomplete and vague. Several and often very different approaches have been used to study the inference systems. When knowledge is represented as a set of facts and rules, this uncertainty can be handled by means of fuzzy logic.

A few years ago, on the beginning of our research a possible combination of Datalog-like languages and fuzzy logic was introduced [1]. In our works the concept of fuzzy Datalog has been introduced by completing the Datalog-rules and facts by an uncertainty level and an implication operator. The level of a rule-head can be inferred from the level of the body and the level of the rule by the implication operator of the rule. Based upon our previous works, later on a fuzzy knowledge-base was developed [2]. In the last years new steps was taken into the direction of multivalued knowledgebase: the fuzzy Datalog was extend to some multivalued direction [3], [4]. In this recent paper the concept of fuzzy-, intuitionistic- and bipolar Datalog will be summarized and for a further step the concept of a possible multivalued knowledgebase will be shown. This knowledge-base is a quadruple of a deduction mechanism; a background knowledge; an algorithm connecting the previous two part; and a decoding set computing the uncertainty levels of the consequences.

II. Multivalued Datalog

Firstly we deal with the deduction mechanism, which is based on Datalog language. Datalog is a logical programming language designed for use as a data-base query language. A Data-log program consists of facts and rules. Using these rules new facts can be inferred from the program's facts. It is very important that the solution of a program be logically correct, that is evaluating the program, the result be a model of the first order logic formulas, being its rules. On the other hand it is also important that this model would contain only those true facts which are the consequences of the program, that is the minimality of this model is expected, i.e. in this model it is impossible to make any true fact false and still have a model consistent with the database.

In the case of Datalog programs there are several equivalent approaches to define the semantics of the program. In fuzzy extension we mainly rely on the fixed-point base aspect.

A. Fuzzy Datalog

In fuzzy Datalog ($fDatalog$) the facts can be completed by an uncertainty level, the rules by an uncertainty level and an implication operator. The level of a rule-head can be infered from the level of the rule-body and the level of the rule by the implication operator of the rule. As in classical cases, the logical correctness is extremely important as well, i.e., the solution would be a model of the program. This means that for each rule of the program, evaluating the fuzzy implication connecting to the rule, its truth-value has to be at least as large as the given uncertainty level. More precisely, the notion of fuzzy rule is the following:

An $fDatalog$ rule is a triplet $r;\beta;I$, where r is a formula of the form $A \leftarrow A_1, ..., A_n (n \geq 0)$, A is an atom (the head of the rule), $A_1, ..., A_n$ are literals (the body of the rule); I is an implication operator and $\beta \in (0, 1]$ (the level of the rule).

For getting a finite result, all the rules in the program must be safe. An $fDatalog$ rule is safe if all variables occurring in the head also occur in the body, and all variables occurring in a negative literal also occur in a positive one. An $fDatalog$ program is a finite set of safe $fDatalog$ rules.

There is a special type of rule, called fact. A fact has the form $A \leftarrow;\beta;I$. From now on, we refer to facts as (A, β), because according to implication I, the level of A can easily be computed. To be short we sometimes denote $\alpha_{A_1 \wedge ... \wedge A_n}$, by α_{body} and α_A by α_{head}.

In the extensions of Datalog several implication operators are used, but in all cases we are restricted to min-max conjunction and disjunction, and to the complement to 1 as negation. So: $\alpha_{A \wedge B} = min(\alpha_A, \alpha_B)$, $\alpha_{A \vee B} = max(\alpha_A, \alpha_B)$ and $\alpha_{\neg A} = 1 - \alpha_A$.

The semantics of $fDatalog$ is defined as the fixed points of consequence transformations. Depending on these transformations, two semantics for $fDatalog$ can be defined [1].

A. Achs is with the Department of Computer Science, Faculty of Engineering University of Pecs, Pecs, Boszorkany u. 2, Hungary e-mail: achs@witch.pmmf.hu

The deterministic semantics is the least fixed point of a deterministic transformation, the nondeterministic semantics is the least fixed point of a nondeterministic transformation. According to the deterministic transformation the rules of a program are evaluated in parallel, while in the nondeterministic case the rules are considered independently and sequentially. Further on because of the lack of space we deal only with the nondeterministic semantics. This is appropriate because in the cases when any rule contains negation, only the nondeterministic semantics is applicable. This transformation is the following:

Let B_P be the Herbrand base of the program P, and let $F(B_P)$ denote the set of all fuzzy sets over B_P. The nondeterministic consequence transformation $NT_P : F(B_P) \to F(B_P)$ is defined as

$$NT_P(X) = \{(A, \alpha_A)\} \cup X$$

where $(A \leftarrow A_1, \ldots, A_n; \beta; I) \in ground(P)$, $(|A_i|, \alpha_{A_i}) \in X$, $1 \leq i \leq n$; $\alpha_A = \max\left(0, \min\left\{\gamma \mid I(\alpha_{body}, \gamma) \geq \beta\right\}\right)$.

There ground(P) is the set of all possible rules of P the variables of which are replaced by ground terms of the Herbrand universe of P. $|A_i|$ denotes the kernel of the literal A_i, (i.e., it is the ground atom A_i, if A_i is a positive literal, and $\neg A_i$, if A_i is negative) and $\alpha_{body} = \min(\alpha_{A_1}, \ldots, \alpha_{A_n})$.

In [1] it is proved that starting from the set of facts, NT_P has a fixed point which is the least fixed points in the case of positive P. This fixed point is denoted by $lfp(NT_P)$. It was also proved, that $lfp(NT_P)$ is a model of P, so $lfp(NT_P)$ could be defined as the nondeterministic semantics of $fDatalog$ programs. If the program P is negation-free then $lfp(NT_P)$ is a minimal model and under certain conditions it is minimal in the case of programs containing any negation as well. These conditions are referred to as stratification. Stratification gives an evaluating sequence in which the negative literals are evaluated first [1].

To compute the level of rule-heads, we need the concept of the uncertainty-level function, which is:

$$f(I, \alpha, \beta) = \min(\{\gamma \mid I(\alpha, \gamma) \geq \beta\}).$$

According to this function the level of a rule-head is: α_{head}= $f(I, \alpha_{body}, \beta)$.

In the former papers several implications were detailed, and the conditions of the existence of an uncertainty-level function was examined for all these operators. For intuitionistic cases three of them was examined till now. They are the following: Gödel (I_G), Lukasiewicz (I_L) and Kleene-Dienes (I_K) operators.

$$\begin{aligned}
I_G(\alpha, \gamma) &= \begin{cases} 1 & \alpha \leq \gamma \\ \gamma & \text{otherwise,} \end{cases} \\
f(I_G, \alpha, \beta) &= \min(\alpha, \beta). \\
I_L(\alpha, \gamma) &= \begin{cases} 1 & \alpha \leq \gamma \\ 1 - \alpha + \gamma & \text{otherwise,} \end{cases} \\
f(I_L, \alpha, \beta) &= \max(0, \alpha + \beta - 1). \\
I_K(\alpha, \gamma) &= \max(1 - \alpha, \gamma), \\
f(I_K, \alpha, \beta) &= \begin{cases} 0 & \alpha + \beta \leq 1 \\ \beta & \alpha + \beta > 1. \end{cases}
\end{aligned}$$

Example 1: Let us consider the next program:

$$\begin{aligned}
&(p(a), 0.8). \\
&(r(b), 0.6). \\
&s(x) \leftarrow q(x, y); 0.7; I_L. \\
&q(x, y) \leftarrow p(x), r(y); 0.7; I_G. \\
&q(x, y) \leftarrow \neg q(y, x); 0.9; I_K.
\end{aligned}$$

As the program has a negation, so according to the stratification the right order of rule-evaluation is 2.,3,.1. Then

$$\begin{aligned}
&lfp(NT_P) = \\
&\{(p(a), 0.8); (r(b), 0.6); (q(a, b), 0.6); \\
&(q(b, a), 0.9); (s(a), 0.3); (s(b), 0.6)\}.
\end{aligned}$$

B. Multivalued extensions of fuzzy Datalog

In fuzzy set theory the membership of an element in a fuzzy set is a single value between zero and one, and the degree of non-membership is automatically just the complement to 1 of the membership degree. However a human being who expresses the degree of membership of a given element in a fuzzy set, very often does not express a corresponding degree of non-membership as its complement. That is, there may be some hesitation degree. This illuminates a well-known psychological fact that linguistic negation does not always correspond to logical negation. Because of this observation, as a generalization of fuzzy sets, the concept of intuitionistic fuzzy sets was introduced by Atanassov in 1983 [5]. In the next paragraphs some possible multivalued extensions will be summarized. (Detailed in [3], [4].)

While in fuzzy logic the uncertainty is represented by a single value (μ), in intuitionistic-(IFS) and interval-valued (IVS) fuzzy logic it is represented by two values, $\vec{\mu} = (\mu_1, \mu_2)$. In the intuitionistic case the two elements must satisfy the condition $0 \leq \mu_1 + \mu_2 \leq 1$, while in the interval-valued case the condition is $0 \leq \mu_1 \leq \mu_2 \leq 1$. In IFS μ_1 is the degree of membership and μ_2 is the degree of non-membership, while in IVS the membership degree is between μ_1 and μ_2. It is obvious that the relations $\mu'_1 = \mu_1$, $\mu'_2 = 1 - \mu_2$ create a mutual connection between the two systems. In both cases an ordering relation can be defined, and according to this ordering a lattice is taking shape:
L_F and L_V are lattices of IFS and IVS respectively, where:

$$\begin{aligned}
&L_F = \{(x_1, x_2) \in [0, 1]^2 \mid x_1 + x_2 \leq 1\}, \\
&\quad (x_1, x_2) \leq_F (y_1, y_2) \Leftrightarrow x_1 \leq y_1, x_2 \geq y_2 \\
&L_V = \{(x_1, x_2) \in [0, 1]^2 \mid x_1 \leq x_2\}, \\
&\quad (x_1, x_2) \leq_V (y_1, y_2) \Leftrightarrow x_1 \leq y_1, x_2 \leq y_2
\end{aligned}$$

It can be proved that both L_F and L_V are complete lattices [6]. Each kind of multivalued $Datalog$ is defined on these lattices and the necessary concepts are generalizations of the ones presented above.

Fuzzy Datalog was extended to intuitionistic and interval-valued direction ($ifDatalog$), and there is a so called bipolar extension of it ($bfDatalog$).

Let $FV(B_P)$ denote the set of all intuitionistic or interval-valued sets over B_P. $ifDatalog$ is a finite set of safe $ifDatalog$ rules $(r;\vec{\beta};\vec{I}_{FV})$; the corresponding consequence transformation $iNT_P : FV(B_P) \to FV(B_P)$ is formally the same as NT_P in (1) except:

$$\vec{\alpha}_A = \max(\vec{0}_{FV}, \min\{\vec{\gamma} \mid \vec{I}_{FV}(\vec{\alpha}_{body}, \vec{\gamma}) \geq_{FV} \vec{\beta}\}).$$

The extended uncertainty-level function is

$$\vec{f}_{FV}(\vec{I}_{FV}, \vec{\alpha}, \vec{\beta}) = \min(\{\vec{\gamma} \mid \vec{I}_{FV}(\vec{\alpha}, \vec{\gamma}) \geq_{FV} \vec{\beta}\}),$$

where $\vec{\alpha}$, $\vec{\beta}$, $\vec{\gamma}$ are elements of L_F, L_V respectively, $\vec{I}_{FV} = \vec{I}_F$ or $\vec{I}_V$ is an implication of L_F or L_V, $\vec{0}_{FV}$ is $\vec{0}_F = (0, 1)$ or $\vec{0}_V = (0, 0)$ and $\geq_{FV}$ is $\geq_F$ or $\geq_V$.

An interpretation is a model, if for each

$$(A \leftarrow A_1, ..., A_n; \vec{\beta}; \vec{I}_{FV}) \in ground(P),$$
$$\vec{I}_{FV}(\vec{\alpha}_{body}, \vec{\alpha}_A) \geq_{FV} \vec{\beta}.$$

In [3] it was proved that this transformation has a least fixed point $lfp(iNT_P)$ which is the least model of the negation-free program P. In $fDatalog$ a fact can be negated by completing its membership degree to 1. In $ifDatalog$ the uncertainty level of a negated fact can be computed according to negators. A negator on L_F or L_V is a decreasing mapping ordering $\vec{0}_{FV}$ and $\vec{1}_{FV}$ together [6]. The applied negators are essential to the computational meaning of a program, but they have no influence on the stratification. So for a stratified $ifDatalog$ program P there is an evaluation sequence in which $lfp(iNT_P)$ is a unique minimal model of P. Therefore $lfp(iNT_P)$ can be considered as the semantics of $ifDatalog$.

The coordinates of intuitionistic and interval-valued implication operators can be determined by each other:

$$I_{V1} = I_{F1}(\vec{\alpha}', \vec{\gamma}'), \qquad \vec{\alpha}' = (\alpha_1, 1 - \alpha_2),$$
$$I_{V2} = 1 - I_{F2}(\vec{\alpha}', \vec{\gamma}')); \qquad \vec{\gamma}' = (\gamma_1, 1 - \gamma_2).$$

The uncertainty-level functions can be computed according to the applied implication:

$$\vec{f}_F(\vec{I}_F, \vec{\alpha}, \vec{\beta}) = (\min(\{\gamma_1 \mid I_{F1}(\vec{\alpha}, \vec{\gamma}) \geq \beta_1\}), \max(\{\gamma_2 \mid I_{F2}(\vec{\alpha}, \vec{\gamma}) \leq \beta_2\}));$$
$$\vec{f}_V(\vec{I}_V, \vec{\alpha}, \vec{\beta}) = (\min(\{\gamma_1 \mid I_{V1}(\vec{\alpha}, \vec{\gamma}) \geq \beta_1\}), \min(\{\gamma_2 \mid I_{V2}(\vec{\alpha}, \vec{\gamma}) \geq \beta_2\})).$$

Based on [6] in [3], [4] four intuitionistic and interval-valued implication was examined, one possible extension of the Kleene-Dienes, one of the Lukasiewicz and two possible extensions of the Gödel implication. Beside determining the appropriate uncertainty-level functions, it was examined whether the consequences of the program remain within the scope of intuitionistic or interval-valued fuzzy logic. That is if the levels of the body and the rule satisfy the conditions referring to intuitionistic or interval-valued concepts, does the resulting level of the head also satisfy these conditions? A further important question is whether the fixed-point algorithm terminates or not, that is whether or not the consequence transformations reach the fixed point in finite steps. Unfortunately in the case of the examined implications with the exception of one of the extensions of the Gödel operator the consequences remain within these scopes only under certain conditions. Maybe in practical cases these conditions are fulfilled. Easy to prove that in the case of this extended Gödel operator the fixed-point algorithm terminates. This operator and the appropriate uncertainty functions are:

$$\vec{I}_{FG}(\vec{\alpha}, \vec{\gamma}) = \begin{cases} (1, 0) & \alpha_1 \leq \gamma_1, \alpha_2 \geq \gamma_2 \\ (\gamma_1, \gamma_2) & \text{otherwise.} \end{cases}$$

$$\vec{I}_{VG}(\vec{\alpha}, \vec{\gamma}) = \begin{cases} (1, 1) & \alpha_1 \leq \gamma_1, \alpha_2 \leq \gamma_2 \\ (\gamma_1, \gamma_2) & \text{otherwise;} \end{cases}$$

$$f_{F1}(\vec{I}_{FG}, \vec{\alpha}, \vec{\beta}) = \min(\alpha_1, \beta_1),$$
$$f_{F2}(\vec{I}_{FG}, \vec{\alpha}, \vec{\beta}) = \max(\alpha_2, \beta_2);$$

$$f_{V1}(\vec{I}_{VG}, \vec{\alpha}, \vec{\beta}) = \min(\alpha_1, \beta_1),$$
$$f_{V2}(\vec{I}_{VG}, \vec{\alpha}, \vec{\beta}) = \min(\alpha_2, \beta_2).$$

The above mentioned problem of extended implications other than G and the results of certain psychological researches have led to the idea of bipolar fuzzy Datalog. The intuitive meaning of intuitionistic degrees is based on psychological observations, namely on the idea that concepts are more naturally approached through separately envisaging positive and negative instances [7], [8], [9]. Taking a further step, there are differences not only in the instances but also in the way of thinking as well. There is a difference between positive and negative thinking, between deducing positive or negative uncertainty. The idea of bipolar Datalog is based on the previous observation: we use two kinds of ordinary fuzzy implications for positive and negative inference, namely we define a pair of consequence transformations instead of a single one. Since in the original transformations lower bounds are used with degrees of uncertainty, therefore starting from IFS facts, the resulting degrees will be lower bounds of membership and non-membership respectively, instead of the upper bound for non-membership. However, if each non-membership value μ is transformed into membership value $\mu' = 1 - \mu$, then both members of head-level can be inferred similarly. Therefore, the appropriate concepts are the following.

The bipolar fDatalog program ($bfDatalog$) is a finite set of safe $bfDatalog$ rules $(r; (\beta_1, \beta_2); (I_1, I_2))$ and the nondeterministic bipolar consequence transformation $b\vec{N}T_P = (NT_{P1}, NT_{P2}) : FV(B_P) \to FV(B_P)$ is similar to NT_P in (1), except in NT_{P2} the level of rule's head is: $\alpha_2' = \max(0, \min\{\gamma_2' | I_2(\alpha_{body2}', \gamma_2') \geq \beta_2'\})$, where $\alpha_{body2}' = \min(\alpha_{A_1 2}', \ldots, \alpha_{A_N 2}'.)$

The uncertainty-level function is: $\vec{f}_b = (f_{b1}, f_{b2})$;

$$f_{b1} = \min\{\gamma_1 \mid I_1(\alpha_1, \gamma_1) \geq \beta_1\};$$
$$f_{b2} = 1 - \min\{1 - \gamma_2 \mid I_2(1 - \alpha_2, 1 - \gamma_2) \geq \ 1 - \beta_2\}.$$

It is evident, that applying the transformation $\mu_1' = \mu_1$, $\mu_2' = 1 - \mu_2$, for each IFS levels of the program, they can be applied to IVS degrees as well. Contrary to the results of $ifDatalog$, the resulting degrees of most variants of bipolar fuzzy Datalog satisfy the conditions referring to IFS and IVS respectively.

The bipolar fDatalog program ($bfDatalog$) is a finite set of safe $bfDatalog$ rules $(r; (\beta_1, \beta_2); (I_1, I_2))$ and the nondeterministic bipolar consequence transformation $b\vec{NT}_P = (NT_{P1}, NT_{P2}) : FV(B_P) \to FV(B_P)$ is similar to NT_P in (1), except in NT_{P2} the level of rule's head is: $\alpha'_2 = \max(0, \min\{\gamma'_2 | I_2(\alpha'_{body2}, \gamma'_2) \geq \beta'_2\})$, where $\alpha'_{body2} = \min(\alpha'_{A_1 2}, \ldots, \alpha'_{A_N 2}.)$

The uncertainty-level function is: $\vec{f_b} = (f_{b1}, f_{b2})$;

$$f_{b1} = \min\{\gamma_1 \mid I_1(\alpha_1, \gamma_1) \geq \beta_1\};$$
$$f_{b2} = 1 - \min\{1 - \gamma_2 \mid I_2(1 - \alpha_2, 1 - \gamma_2) \geq 1 - \beta_2\}.$$

It is evident, that applying the transformation $\mu'_1 = \mu_1$, $\mu'_2 = 1 - \mu_2$, for each IFS levels of the program, they can be applied to IVS degrees as well. Contrary to the results of $ifDatalog$, the resulting degrees of most variants of bipolar fuzzy Datalog satisfy the conditions referring to IFS and IVS respectively.

Example 2: Consider the next IFS valued program:

$$(p(a), (0.6, 0.25)). \quad (q(a), (0.7, 0.1)). \quad (r(a), (0.7, 0.3)).$$
$$q(x) \leftarrow p(x), r(x); (0.8, 0.15); \vec{I}.$$

Let $\vec{I} = \vec{I}_{FG}$, then $\vec{\alpha}_{body} = \min((0.6, 0.25), (0.7, 0.3)) = (0.6, 0.3)$, $f_1(I_{FG}, \vec{\alpha}, \vec{\beta}) = \min(\alpha_1, \beta_1) = 0.6$; $f_2(I_{FG}, \vec{\alpha}, \vec{\beta}) = \max(\alpha_2, \beta_2) = 0.3$, that is the level of rule's head is (0.6, 0.3). Allowing the other levels of q(a), its resulting levels are the union of them: $\max((0.8, 0.3), (0.7, 0.1)) = (0.8, 0.1)$. So the fixed point of the program is:

$$\{(p(a), (0.6, 0.25)), (r(a), (0.7, 0.3)), (q(a), (0.8, 0.1))\}$$

Now let the program be evaluated in bipolar manner and let $I = (I_L, I_G)$. Then $\alpha_{body1} = \min(0.6, 0.7) = 0.6$, $\alpha'_{body2} = \min(1 - 0.3, 1 - 0.25) = 0.7$; $f_{b1}(I_L, \alpha_1, \beta_1) = \max(0, \alpha_1 + \beta_1 - 1) = 0.8 + 0.6 - 1 = 0.4$; $f_{b2}(I_G, \alpha'_2, \beta'_2) = 1 - \min(\alpha'_2, 1 - \beta_2) = 1 - \min(0.7, 0.85) = 0.3$. Allowing the other levels of q(a), its resulting levels are $(\max(0.4, 0.7), 1 - \max(1 - 0.3, 1 - 0.1)) = (0.7, 0.1)$, so the fixed point is:

$$\{(p(a), (0.6, 0.25)), (r(a), (0.7, 0.3)), (q(a), (0.7, 0.1))\}.$$

As fuzzy Datalog is a special kind of its each multivalued extension, so further on both fDatalog and any of above extensions will be called multivalued Datalog ($mDatalog$).

III. Background Knowledge

The facts and rules of a $mDatalog$ program can be regarded as any kind of knowledge, but sometimes we need some other information in order to get an answer for a query. In this section we give a possible model of background knowledge. Some kind of synomyms will be defined between the potential predicates and between the potential constans of the given problem, so it can be examined in a larger context. More precisely a proximity relation will be defined between predicates and between constants and these structures of proximity will serve as a background knowledge.

Definition 1: A multivalued proximity on a domain D is an IFS or IVS valued relation $\vec{R}_{FV_D} : D \times D \to [\vec{0}_{FV}, \vec{1}_{FV}]$ which satisfies the following properties:

$$\vec{R}_{F_D}(x, y) = \vec{\mu}_F(x, y) = (\mu_1, \mu_2), \quad \mu_1 + \mu_2 \leq 1$$
$$\vec{R}_{V_D}(x, y) = \vec{\mu}_V(x, y) = (\mu_1, \mu_2), \quad 0 \leq \mu_1 \leq \mu_2 \leq 1$$
$$\vec{R}_{FV_D}(x, x) = \vec{1}_{FV} \quad \forall x \in D \quad (reflexivity)$$
$$\vec{R}_{FV_D}(x, y) = \vec{R}_{FV_D}(y, x) \quad \forall x, y \in D \quad (symmetry).$$

A proximity is similarity if it is transitive, that is

$$\vec{R}_{FV_D}(x, z) \geq \min(\vec{R}_{FV_D}(x, y), \vec{R}_{FV_D}(y, z)) \; \forall x, y, z \in D.$$

In the case of similarity equivalence classifications can be defined over D allowing to develop simpler or more effective algorithms, but now we deal with the more general proximity.

In our model the background knowledge is a set of proximity sets.

Definition 2: Let $d \in D$ any element of domain D. The proximity set of d is an IFS or IVS subset over D:

$$R_{FV_d} = \{(d_1, \vec{\lambda}_{FV_1}), (d_2, \vec{\lambda}_{FV_2}), \ldots, (d_n, \vec{\lambda}_{FV_n})\},$$

where $d_i \in D$ and $\vec{R}_{FV_D}(d, d_i) = \vec{\lambda}_{FV_i}$ for i = 1,..., n.

Based on proximities a background knowledge can be constructed which means some information about the proximity of terms and predicate symbols.

Definition 3: Let G be any set of ground terms and S any set of predicate symbols. Let RG_{FV} and RS_{FV} be any proximity over G and S respectively. The background knowledge is:

$$Bk = \{RG_{FV_t} \mid t \in G\} \cup \{RS_{FV_p} \mid p \in S\}$$

IV. Multivalued Knowledge-Base

So far two steps was made on the way leading to the concept of multivalued knowledge-base: the concept of a multivalued Datalog program and the concept of background knowledge was defined. Now the question is: how can we connect this program with the background knowledge? How can we deduce to the "synonyms"? For example if $(r(a), (0.8, 0.1))$ is an IFS fact and $RS_F(r, s) = (0.6, 0.3), RG_F(a, b) = (0.7, 0.2)$ then what is the uncertainty of $r(b), s(a)$ or $s(b)$?

To solve this problem the concept of extended uncertainty function will be introduced. According to this function the uncertainty levels of synonyms can be computed from the levels of original fact and from the proximity values of actual predicates and its arguments. It is expectable that in the case of identity the level must be unchanged, but in other cases it is to be less or equal then the original level or then the proximity values. Furthermore we require this function to be monotonically increasing. This function will be ordered to each atom of a program.

Let p be a predicate symbol with n arguments, then p/n is called the functor of the atom characterized by this predicate symbol.

Definition 4: An extended uncertainty function of p/n is:

$$\vec{\varphi}_p(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) : (\vec{0}_{FV}, \vec{1}_{FV}]^{n+2} \rightarrow [\vec{0}_{FV}, \vec{1}_{FV}]$$

where

$$\vec{\varphi}_p(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) \leq \min(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}),$$
$$\vec{\varphi}_p(\vec{\alpha}, \vec{1}_{FV}, \vec{1}_{FV}, \ldots, \vec{1}_{FV}) = \vec{\alpha}$$

and $\vec{\varphi}_p(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n})$ is monoton increasing in each argument.

It is worth mentioning that any triangular norm is suitable for extended uncertainty function, for example

$$\vec{\varphi}_{p_1}(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) = \min(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}),$$
$$\vec{\varphi}_{p_2}(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) = \min(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1} \cdots \vec{\lambda_n}),$$

where the product is:

$$(\mu_1, \mu_2) \cdot (\lambda_1, \lambda_2) = (\mu_1 \cdot \lambda_1, \mu_2 \cdot \lambda_2)$$

are extended uncertainty functions, but

$$\vec{\varphi}_{p_3}(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) = \vec{\alpha} \cdot \vec{\lambda} \cdot \vec{\lambda_1} \cdots \vec{\lambda_n}$$

is an extended uncertainty function only in the interval valued case.

Example 3: Let $(r(a), (0.8, 0.1))$ be an IFS fact and $RS_F(r, s) = (0.6, 0.3), RG_F(a, b) = (0.7, 0.2)$ and $\vec{\varphi}_r(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}) = \min(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1})$ then the uncertainty levels of $r(b), s(a)$ and $s(b)$ are:

$$(r(b), (\min(0.8, 1, 0.7), \max(0.1, 0, 0.2))) = (r(b), (0.7, 0.2))$$
$$(s(a), (\min(0.8, 0.6, 1), \max(0.1, 0.3, 0))) = (s(a), (0.6, 0.3))$$
$$(s(b), (\min(0.8, 0.6, 0.7), \max(0.1, 0.3, 0.2))) = (s(b), (0.6, 0.3))$$

We have to order extended uncertainty functions to each predicate of the program. The set of these functions will be the function-set of the program.

Definition 5: Let P be a multivalued Datalog program, and F_P be the set of the program's functors. The function-set of P is:

$$\Phi_P = \{\vec{\varphi}_p(\vec{\alpha}, \vec{\lambda}, \vec{\lambda_1}, \ldots, \vec{\lambda_n}) \mid \forall\, p/n \in F_P\}$$

Let P be a multivalued Datalog program, Bk be any background knowledge and Φ_P be the function-set of P. The deducing mechanism consist of two alternating part: starting from the fact we determine their "synonyms", then applying the suitable rules another facts are derived, then their "synonyms" are derived and again the rules are applied, etc. To define it in a precise manner the concept of modified consecution transformation will be introduced.

The original consequence transformation is defined over the set of all multivalued sets of P's Herbrand base, that is over $F(B_P)$. To define the modified transformation's domain, let us extend P's Herbrand universe with all possible ground terms occurring in background knowledge: this way, we obtain the modified Herbrand universe $modH_P$. Let the modified Herbrand base $modB_P$ be the set of all possible ground atoms whose predicate symbols occur in $P \cup Bk$ and whose arguments are elements of $modH_P$. This leads to

Definition 6: The modified consequence transformation

$$modNT_P : FV(modB_P) \rightarrow FV(modB_P)$$

is defined as

$$\begin{aligned} modNT_P(X) = \{(q(s_1, ..., s_n), \vec{\varphi}_p(\vec{\alpha_p}, \vec{\lambda}_q, \vec{\lambda}_{s_1}, \ldots, \vec{\lambda}_{s_n}) \mid \\ (q, \vec{\lambda}_q) \in RS_{FV_p}; \\ (s_i, \vec{\lambda}_{s_i}) \in RG_{t_i},\ 1 \leq i \leq n\} \cup X, \end{aligned}$$

where

$(p(t_1, ..., t_n) \leftarrow A_1, \ldots, A_k; \vec{I}; \vec{\beta}) \in \text{ground}(P)$,
$(\,|A_i|, \alpha_{A_i}) \in X,\ 1 \leq i \leq k,\quad (|A_i|$ is the kernel of $A_i)$

and $\vec{\alpha_p}$ is computed according to the actual extension of (1).

It is obvious that this transformation is inflationary over $FV(modB_P)$ and it is monotone if P is positive.
(A transformation T over a lattice L is inflationary if $X \leq T(X)\ \forall X \in L$. T is monotone if $T(X) \leq T(Y)$ if $X \leq Y$.)

According to [10] an inflationary transformation over a complete lattice has a fixed point moreover a monotone transformation has a least fixed point, so

Proposition 1: The modified consequence transformation $modNT_P$ has a fixed point. If P is positive, then this is the least fixed point.

It can be shown that this fixed point is a model of P, but $lfp(NT_P) \subseteq lfp(modNT_P)$, so it is not a minimal model.

As the modifying of original transformation that is the modifyingalgorithm has no effect on the order of rules, therefore it does not change the stratification. Therefore we can state

Proposition 2: In the case of stratified program P, $modNT_P$ has least fixed point as well.

Now we have all components together to define the concept of a multivalued knowledge-base. But before doing it, it is worth mentioning that the above modified consequence transformation is not the unique way to connect the background knowledge with the deduction mechanism, there could be other possibilities as well.

Definition 7: A multivalued knowledge-base (mKB) is a quadruple

$$mKB = (Bk, P, \Phi_P, cA),$$

where Bk is a background knowledge, P is a multivalued Datalog program, Φ_P is a function-set of P and cA is any connecting algorithm.

The result of the connected and evaluated program is called the consequence of the knowledge-base, denoted by

$$C(Bk, P, \Phi_P, cA).$$

So in our case $C(Bk, P, \Phi_P, cA) = lfp(modNT_P)$.

Example 4: Let the IVS valued $mDATALOG$ program and the background knowledge be as follows

$$lo(x, y) \leftarrow gc(y), mu(x); (0.7, 0.9); \vec{I}_{VG}.$$
$$(fv(V), (0.85, 0.9).$$
$$(mf(M), (0.7, 0.8).$$

	B	V	M
B	$(1,1)$	$(0.8,0.9)$	
V	$(0.8,0.9)$	$(1,1)$	
M			$(1,1)$

	lo	li	gc	fv	mu	mf
lo	$(1,1)$	$(0.7,0.9)$				
li	$(0.7,0.9)$	$(1,1)$				
gc			$(1,1)$	$(0.8,0.9)$		
fv			$(0.8,0.9)$	$(1,1)$		
mu					$(1,1)$	$(0.6,0.7)$
mf					$(0.6,0.7)$	$(1,1)$

According to the connecting algorithm, it is enough to consider only the extended uncertainty functions of head-predicates. Let these functions be as follows:

$$\vec{\varphi}_{lo}(\vec{\alpha},\vec{\lambda},\vec{\lambda_1},\vec{\lambda_2}) := \min(\vec{\alpha},\vec{\lambda},\vec{\lambda_1}\cdot\vec{\lambda_2}),$$
$$\vec{\varphi}_{fv}(\vec{\alpha},\vec{\lambda},\vec{\lambda_1}) := \min(\vec{\alpha},\vec{\lambda},\vec{\lambda_1}),$$
$$\vec{\varphi}_{mf}(\vec{\alpha},\vec{\lambda},\vec{\lambda_1}) := \vec{\alpha}\cdot\vec{\lambda}\cdot\vec{\lambda_1}.$$

The modified consequence transformation takes shape in the following steps:

$$X_0 = \{(fv(V),(0.85,0.9)),(mf(M),(0.7,0.8))\}$$

$\Downarrow$ (according to the proximity)

$$X_1 = modNT_P(X_0) = X_0 \cup$$
$$\{(gc(V),\vec{\varphi}_{fv}((0.85,0.9),(0.8,0.95),(1,1)) =$$
$$(min(0.85,0.8,1),min(0.9,0.95,1)) = (0.8,0.95)),$$
$$(fv(B),\vec{\varphi}_{fv}((0.85,0.9),(1,1),(0.8,0.9)) = (0.8,0.9)),$$
$$(gc(B),\vec{\varphi}_{fv}((0.85,0.9),(0.8,0.95),(0.8,0.9)) = (0.8,0.9)),$$
$$(mu(M),\vec{\varphi}_{mf}((0.7,0.8),(0.6,0.7),(1,1)) =$$
$$(0.7\cdot 0.6\cdot 1, 0.8\cdot 0.7\cdot 1) = (0.42,0.56))\}$$

$\Downarrow$ (applying the rules)

$$lo(M,V) \leftarrow gc(V),mu(M);(0.7,0.9);\vec{I}_{VG}.$$
$$lo(M,B) \leftarrow gc(B),mu(M);(0.7,0.9);\vec{I}_{VG}.$$

here : $f_V(\vec{I}_{VG},\vec{\alpha},\vec{\beta}) = \min(\vec{\alpha}_{body},\vec{\beta})$, so

$$X_2 = modNT_P(X_1) = X_1 \cup$$
$$\{(lo(M,V),(0.42,0.56)),(lo(M,V),(0.42,0.56))\}$$

$\Downarrow$ (according to the proximity)

$$X_3 = modNT_P(X_2) = X_2 \cup$$
$$\{(li(M,V),(min(0.42,0.7,1\cdot 1),min(0.56,0.9,1\cdot 1))),$$
$$(li(M,B),(min(0.42,0.7,1\cdot 1),min(0.56,0.9,1\cdot 1)))\}\cup$$
$$\{(li(M,V),(min(0.42,0.7,0.64),min(0.56,0.9,0.81))),$$
$$(li(M,B),(min(0.42,0.7,0.64),min(0.56,0.9,0.81)))\}$$

X_3 is a fixed point, so the consequence of the knowledge-base is:

$$C(Bk,P,\Phi_P,cA) =$$
$$\{(fv(V),(0.85,0.9)),(mf(M),(0.7,0.8)),$$
$$(gc(V),(0.8,0.95)),(fv(B),(0.8,0.9)),$$
$$(gc(B),(0.8,0.9)),(mu(M),(0.42,0.56)),$$
$$(lo(M,V),(0.42,0.56)),(lo(M,V),(0.42,0.56))$$
$$(li(M,V),(0.42,0.56)),(li(M,B),(0.42,0.56))\}$$

To illustrate our discussion with some realistic content, in the above example the knowledge-base could have the following interpretation. Let us suppose that music listeners "generally"(level between 0.7, 0.9) are fond of the greatest composers. Assume furthermore that Mary is a "rather devoted" (level between 0.7, 0.8) fan of classical music (mf), and Vivaldi is "generally accepted" (level between 0.85, 0.9) as a "great composer". It is also widely accepted that the music of Vivaldi and Bach are fairly "similar", being related in overall structure and style. On the basis of the above information, how strongly state that Mary likes Bach? To continue with this idea, next we can assume that an internet agent wants to suggest a good CD for Mary, based on her interests revealed through her actions at an internet site. A multivalued knowledge-base could help the agent to get a good answer. As some of the readers may well know, similar mechanisms – but possibly based on entirely different modelling paradigms – are in place in prominent websites such as Amazon and others.

V. Conclusions

In this paper a possible model of multivalued knowledge-base was introduced. The knowledge-handling of the model is based on multivalued Datalog as a deduction mechanism and multivalued proximity serving for handling some "synonyms". Supposedly there is a lot of possibility for developing this model. For example to work out an efficient evaluating algorithm or to find another connection methods combinig the background knowledge and the deduction mechanism to each other.

References

[1] A. Achs and A. Kiss, *Fuzzy extension of Datalog*, Acta Cybernetica Szeged, 1995, vol.12., pp. 153–166

[2] A. Achs, *Computed Answer from Uncertain Knowledge: A Model for Handling Uncertain Information*, Computing and Informatics, 2007, vol.26., pp. 63–76

[3] A. Achs, *From Fuzzy- to Bipolar- Datalog*, Proceedings of the 5th EUSFLAT Conference, Ostrava, Czech Republic, September 11-14, 2007, pp. 221–227

[4] A. Achs,*DATALOG-based uncertainty-handling*, The Twelfth IASTED International Conference on Artificial Intelligence and Soft Computing, September 1-3, 2008, pp. 44–49

[5] K. Atanassov, *Intuitionistic Fuzzy Sets*, Springer–Verlag, Heidelberg, 1999.

[6] C. Cornelis and G. Deschrijver and E.E. Kerre, *Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, classification, application*, International Journal of Approximate Reasoning, 2004, vol.35., pp. 55–95

[7] J.T. Cacioppo and W.L. Gardner and G.G. Berntson, *Beyond bipolar conceptualization and measures: the case of attitudes and evaluative spaces*, Personality and Social Psychol, 1997, Rev.1, pp. 3–25

[8] D. Dubois and P. Hajek and H. Prade, *Knowledge-Driven versus Data-Driven Logics*, Journal of Logic, Language, and Information, 2000, vol.9., pp. 65–89

[9] D. Dubois and S. Gottwald and P. Hajek and J. Kacprzyk and H. Prade, *Terminological difficulties in fuzzy set theory The case of Intuitionistic Fuzzy Sets*, Fuzzy Sets and Systems, 2005, vol.15., pp. 485–491

[10] S. Ceri and G. Gottlob and L. Tanca, *Logic Programming and Databases*, Springer-Verlag, Berlin, 1990.