A multi context-based Approximate Reasoning

Luciano Blandi and Maria I. Sessa

Dipartimento Matematica e Informatica
Universita' di Salerno
via Ponte Don Melillo - 84084 Fisciano (SA), Italy
{lblandi, misessa}@unisa.it

Abstract. The paper concerns with the theory of similarity relations in the framework of Logic Programming. Similarity relation [34] is a formal notion that allows us to manage alternative instances of entities that can be considered 'equal' with a given degree. In [8, 15, 30, 29, 18] this notion has been encapsulated in an inference engine, based on the Logic Programming paradigm, modifying the inference model to provide a more flexible unification (approximately equal) than the dichotomic match (equal or not). In many situations, more than one similarity relation can be defined in a universe by taking into account different contexts. In this paper we studied some aggregations of such fuzzy relations and their application to our extented Logic Programming framework.

1 Introduction and previous works

The main interest in Logic Programming field traditionally concerns problems related to the analysis and the efficiency of exact inferences allowed by logic programs [1, 14]. However, very often the need of methods to enhance this capability, in order to deal with approximate information or flexible inference schemes, arises in many applications. In general, approximate reasoning capabilities are introduced in the Logic Programming framework by considering the inference system based on fuzzy logic rather than on conventional two-valued logic.

In [8] a methodology that allows to manage uncertain and imprecise information in the frame of the declarative paradigm of Logic Programming has been proposed. With this aim, a Similarity relation R between function and predicate symbols in the language of a logic program is considered. Approximate inferences are then possible since Similarity relation allows us to manage alternative instances of entities that can be considered "equal" with a given degree.

With respect to the previous literature [4, 31, 12, 13], this approach is very different since the approximation is represented and managed at a syntactic-level, instead of at a rule-level. Roughly speaking, the basic idea is that the fuzziness feature is provided by an abstraction process which exploits a formal representation of similarity relations between elements in the alphabet of the language (constants, functions, predicates). On the contrary, in the underground logic theory, the inference rule as well as the usual crisp representation of the considered universe are not modified. It allows us to avoid both the introduction of weights on the clauses, and the use of fuzzy sets as elements of the language.

In [29] the operational counterpart of this extension is faced by introducing a modified SLD Resolution procedure. Such a procedure allows us to compute numeric values belonging to the interval [0,1] providing an approximation measure of the obtained solutions. These numeric values are computed through a generalized unification mechanism. In [18] a Prolog interpreter written in Java which implements this Similarity-based extension has been presented.

In these works, the approximation was based on a single similarity relation. However, by taking into account different contexts (i.e. points of view), it is possible to define different similarity relations in a given universe. Let us consider the following Example

Example 1.

Let U be a set of animals denoted with

$U = \{M,\ B,\ G,\ E,\ P,\ H,\ S,\ T,\ C,\ W,\ D\}$

where these letters stand for

$M = man,\ \ B = bear,\ \ G = gorilla,\ \ E = eagle,\ P = pigeon,\ \ H = hawk,$

$S = shark,\ \ T = tiger,\ C = cat,\ \ W = wolf,\ \ D = dog.$

We can define a similarity R_1 between elements in U based on feature 'morphology' by setting for any $x, y \in U$

$$R_1(x,y) = R_1(y,x)$$
$$R_1(x,y) = 1 \quad if\ \ x = y$$
$$R_1(G,M) = R_1(D,W) = R_1(E,H) = .8$$
$$R_1(W,T) = R_1(D,T) = .6$$
$$R_1(C,T) = R_1(C,W) = R_1(C,D) = .4$$
$$R_1(B,T) = R_1(B,C) = R_1(B,W) = R_1(B,D) = R_1(E,P) = R_1(P,H) = .2$$
$$R_1(x,y) = 0 \quad otherwise.$$

Also, we can define a similarity R_2 between elements in U based on feature 'aggressiveness' by setting for any $x, y \in U$

$$R_2(x,y) = R_2(y,x)$$
$$R_2(x,y) = 1 \quad if\ \ x = y$$
$$R_2(G,W) = R_2(T,S) = R_2(E,H) = .8$$
$$R_2(M,D) = R_2(B,S) = R_2(B,T) = R_2(C,D) = .6$$
$$\begin{aligned} &R_2(M,E) = R_2(M,H) = R_2(M,C) = R_2(B,G) = R_2(B,W) = R_2(G,S) = \\ &\quad = R_2(G,T) = R_2(E,C) = R_2(E,D) = R_2(H,C) = R_2(H,D) = R_2(S,W) = R_2(T,W) = .4 \end{aligned}$$
$$\begin{aligned} &R_2(M,B) = R_2(M,G) = R_2(M,S) = R_2(M,T) = R_2(W,M) = R_2(B,E) = \\ &\quad = R_2(B,H) = R_2(B,C) = R_2(B,D) = R_2(G,E) = R_2(G,H) = R_2(G,C) = R_2(G,D) = \\ &\quad = R_2(E,S) = R_2(E,T) = R_2(E,W) = R_2(H,S) = R_2(H,T) = R_2(H,W) = R_2(S,C) = \\ &\quad = R_2(S,D) = R_2(T,C) = R_2(T,D) = R_2(C,W) = R_2(W,D) = .2 \end{aligned}$$
$$R_2(x,y) = 0 \quad otherwise.$$

In many cases, we may need to aggregate different relations. Aggregation of binary (fuzzy) relations is an important and challenging mathematical problem in applied areas as social choice, group choice, multiple-criteria decision-making, synthesis of implication functions, etc. Formally, this problem can be formulated in terms of group choice theory as follows: suppose U is a finite set of alternatives and $\mathcal{R} =< R_1, ..., R_n >$ is an ordered n-tuple of binary (fuzzy) relations on U. Elements of $\mathcal{R}$ are regarded as individual preferences and $\mathcal{R}$ is called a *profile* of individual preferences on the set U of alternatives. For a given U, an aggregation rule assigns a group preference R to each profile $\mathcal{R}$ of individual preferences on U (very often it is assumed that $n > 2$). We denote this rule by the same letter R. Depending on the application area, various restrictions are imposed on R. In [5, 11, 2, 26, 27, 21, 28], the fuzzy binary relations $R_1, ..., R_n$ and R satisfy T-transitivity property in which T is an Archimedian t-norm. In our framework we consider transitivity based on the minimum triangular norm. In particular, in this paper we study properties of $R_{min} = \bigcap_i R_i$ and $R_{max} = \bigcup_i R_i$ assuming that individual and group preferences are similarity relations on a finite set of alternative U.

The paper is organized as follows. After preliminaries on similarity relation in Logic Programming framework, Section 3 will study the aggregation of similarity relations by intersection (Subsection 3.1) and union (Subsection 3.2), and their exploitation in the similarity based Logic Programming (Subsection 3.3). The last section contains some concluding remarks.

2 Preliminaries

2.1 Similarity relation

An important and very intuitive theoretical basis for fuzzy subsets is given by the concept of fuzzy equivalence relations, which, in some sense, measure the degree to which two points of the universe are indistinguishable, and which generalize and relax the concept of classical equivalence relations. A strong motivation for this notion follows from the so-called Poincaré Paradox [25]: if for three (real world) objects A, B and C we know that A is indistinguishable from B and B is indistinguishable from C, we cannot necessarily conclude that A is indistinguishable from C too. Fuzzy equivalence relations have been introduced under the name of similarity relation in [34] (with respect to the minimum T_M, the generalization to t-norms was considered in [32]). In this section, we will recall some well-known definitions and properties related to similarity relation and to its application in the Logic Programming framework.

In Cantorian set theory, a relation on a universe U can be identified with a subset of U^2. By analogy, a fuzzy relation on U is then a fuzzy subset of U^2. For early traces of properties of fuzzy relations see [34] and [22–24], more recent treatments include [6, 3].

At first, let us recall that a *T-norm* is a binary operation $\wedge : [0,1] \times [0,1] \to [0,1]$ associative, commutative, non-decreasing in both the variables, and such that $x \wedge 1 = 1 \wedge x = x$ for any x in [0,1]. In the sequel, we assume that $x \wedge y$ is the *minimum* between the two elements $x, y \in [0,1]$.

Definition 1. *Given a T-norm, a fuzzy relation R on a set U is* T-transitive *if and only if $T(R(x,y), R(y,z)) \leq R(x,z)$ for any $x, y, z \in U$.*

Among all T-transitive fuzzy relations, similarity relations and fuzzy T-preorders are the most important ones.

Definition 2. *A* similarity *on a domain U is a fuzzy relation $R : U \times U \to [0,1]$ in U such that the following properties hold*

i) $R(x,x) = 1$ *for any $x \in U$* *(reflexivity)*

ii) $R(x,y) = R(y,x)$ *for any $x, y \in U$* *(symmetry)*

iii) $R(x,z) \geq R(x,y) \wedge R(y,z)$ *for any $x, y, z \in U$* *(transitivity)*

we say that R is strict *if the following implication is also verified*

iv) $R(x,z) = 1 \implies x = z.$

The value $R(x,z)$ can be interpreted as the degree of equality or the degree of indistinguishability of x and y or, equivalently, as the truth value of the statement 'x is equal to y'. The $\wedge$-transitivity is a many-valued model of the proposition 'IF x is equal to y AND y is equal to z THEN x is equal to z'.

Similarities also are called indistinguishability operators, fuzzy equalities, fuzzy equivalences, likeness, probabilistic relations, proximity relations, M-valued equality, depending on the authors and on the t-norm used to model their transitivity.

There is a lot of work around this concept and it has been proved to be a useful tool both in the theoretical aspects of fuzzy logic and in their applications such as fuzzy control or approximate reasoning.

2.2 Similarity relation and closure operators

We synthetically give some well known notions concerning closure operators and equivalence relations.

Definition 3. *Let $(P, \preceq)$ be a poset. An operator $H : P \rightarrow P$ is called a* closure *(resp.* reductive*) operator if for any x, y in P the following properties hold:*

i) $x \preceq H(x) \qquad$ *(resp.* $H(x) \preceq x$*)*

ii) $H(H(x)) = H(x)$

iii) $x \preceq y \Longrightarrow H(x) \preceq H(y)$.

Proposition 1. *Let $\equiv$ be an equivalence relation on a set S and $\mathcal{P}(S)$ the powerset of S. Then, the operator $H_{\equiv} : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ such that for any $X \subseteq \mathcal{P}(S)$*

$$H_{\equiv}(X) = \{x' \in S \quad | \quad \exists x \in X : \quad x' \equiv x)\}$$

is a closure operator.

The following notion of $\lambda-cut$ is crucial in fuzzy set theory:

Definition 4. *Let U be a domain and $R : U \times U \rightarrow [0, 1]$ a fuzzy relation in U. Then, for any $\lambda \in [0, 1]$, the relation $\approxeq_{R,\lambda}$ in U defined as*

$$x \approxeq_{R,\lambda} y \qquad \Longleftrightarrow \qquad R(x, y) \succeq \lambda$$

is named cut of level λ *(in short* λ-cut*) of R.*

Similarity relations are strictly related with equivalence relations and, then, to closure operators. Indeed, the notion of $\lambda - cut$ allows us to define a similarity by means of a suitable family of equivalence relations according to the following result that can be easily proven.

Proposition 2. *Let U be a domain and $R : U \times U \rightarrow [0, 1]$ a Similarity in U. Then, for any $\lambda \in [0, 1]$, the relation $\approxeq_{R,\lambda}$ in U is an equivalence relation. Also, the operator $H_{\approxeq_{R,\lambda}} : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$ such that for any $X \in \mathcal{P}(U)$*

$$H_{\approxeq_{R,\lambda}}(X) = \{z \in U \mid \exists x \in X : \ x \approxeq_{R,\lambda} y\} = \{z \in U \mid \exists x \in X : \ R(z, x) \geq \lambda\},$$

is a closure operator.

Proposition 3. *Let R be a similarity in a domain U and, for any $\lambda \in [0, 1]$ let $\approxeq_{R,\lambda}$ be the $\lambda-cut$ of R. Then, $\{\approxeq_{R,\lambda}\}_{\lambda \in [0,1]}$ is a family of equivalence relations such that,*

i) for any μ and λ in $[0, 1]$, $\qquad \lambda \preceq \mu \quad \Rightarrow \quad \approxeq_{R,\lambda} \supseteq \approxeq_{R,\mu}$

ii) for any μ in $[0, 1]$, $\qquad \bigcap_{\lambda \preceq \mu} \approxeq_{R,\lambda} = \approxeq_{R,\mu}$.

Conversely, let $\{\approxeq_{\lambda}\}_{\lambda \in [0,1]}$ be a family of equivalence relations satisfying conditions i) and ii). Then the relation R defined by setting

$$R(x, y) = Sup\{\lambda \in [0, 1] \quad | \quad x \approxeq_{\lambda} y\}$$

is a similarity whose family of $\lambda-cuts$ is equal to the family $\{\approxeq_{\lambda}\}_{\lambda \in [0,1]}$.

Any $\lambda-$cut can be considered as a generalization of the equality. This notion plays an important rule in our approach. Indeed, the relation $\approxeq_{R,\lambda}$ formalizes the idea that two constant symbols can be considered equal with a fixed approximation level $\lambda \in [0, 1]$. Such a level provides a measure of the allowed approximation in order to avoid failure of matching between constant symbols in a SLD-derivation process.

2.3 Logic Programming with Similarity

We briefly recall that a logic program P is a set of universally quantified Horn clauses on a first order language L, denoted with $H \leftarrow B_1, \ldots, B_k$, and a goal is a negative clause, denoted with $A_1, \ldots, A_n$. We denote with B_L the set of ground atomic formulae in L, i.e. the Herbrand base of L, and with T_P the immediate consequence operator $T_P : \mathcal{P}(B_L) \mapsto \mathcal{P}(B_L)$ defined by:

$$T_P(X) = \{a | a \leftarrow a_1, \ldots, a_n \in \mathit{\Gamma}(P) \text{ and } a_i \in X, 1 \leq i \leq n\}$$

where $\mathit{\Gamma}(P)$ denotes the set of all ground instances of clauses in P. The application of Tarski's fixpoint theorem yields a characterization of the semantics of P, which is the least Herbrand model M_P of P given by:

$$M_P = lfp(T_P) = \bigcup_{n \geq 0} T_P^n(\emptyset)$$

where lfp stands for least fixpoint [1].

In the classical Logic Programming, function and predicate symbols of the language L are crisp elements, i.e., distinct elements represent distinct information and no matching is possible. In [8] the exact matching between different entities is relaxed by introducing a Similarity relation R in the set of constant, function and predicate symbols in the language of a logic program P. In order to deal with the approximation introduced by a similarity relation R, the program P is extended by adding new clauses which are "similar" at least with a fixed degree λ in (0,1] to the given ones. This program transformation is obtained by considering the closure operator H_λ associated to R. The new logic program

$$H_\lambda(\mathit{\Gamma}(P)) = \{C' \in L | \exists C \in \mathit{\Gamma}(P) \text{ such that } R(C, C') \geq \lambda\},$$

named *extended-program of level* λ, allows us to enhance the inference process.

An alternative way to manage the information carried on by the Similarity introduced between function and predicate symbols in P, is given by considering as an unique element different symbols which have Similarity degree greater or equal to λ. In other words, we consider the quotient set of $\approxeq_{R,\lambda}$ as a new alphabet L_λ, where $F/ \approxeq_{R,\lambda}$ and $R/ \approxeq_{R,\lambda}$ are the sets of function and predicate symbols, respectively. More formally, let us denote with $[s] \in L_\lambda$ the equivalence class of a symbol $s \in F \cup R$ with respect to $\approxeq_{R,\lambda}$.We call *translation up* to $\approxeq_{R,\lambda}$ the function:

$$\tau_\lambda : F \cup R \mapsto F/ \approxeq_{R,\lambda} \cup R/ \approxeq_{R,\lambda}$$

defined by setting:

$$\tau_\lambda(x) = x \text{ for any variable } x \in V, \text{ and } \tau_\lambda(f) = [f]$$

for any function/predicate symbol $f \in F \cup R$. Recursively, we can easily define the extension of τ_λ to the sets of formulae in L. Let us consider a logic program P on the language L.

The set

$$P_\lambda = \tau_\lambda(\mathit{\Gamma}(P)) = \{C' \in L_\lambda | C' = \tau_\lambda(C), \;\; C \;\; clause \;\; in \;\; \mathit{\Gamma}(P)\}$$

is a logic program that we name *abstract-program* of level λ.

By considering the abstract program P_λ, it is possible to express information provided by the similarity relation in a syntectic way exploiting the quotient language L_λ. Then, P_λ could be used to manage similarity-based reasoning as well as $H_\lambda(\mathit{\Gamma}(P))$. In [30] the equivalence of these two approaches has been shown for first order languages by using an abstract interpretation technique.

It is worth to stress that the introduced similarity generally changes the semantic of the original program. Indeed, it allows us to add new clauses to P providing the extended program $H_\lambda(P)$. This is the more straight way to implement the approximated inference process based on similarity.

On the other hand, by considering the abstract program P_λ, it is possible to express information provided by the similarity relation in a syntectic way exploiting the quotient language L_λ.

Both these programs allow to perform approximate inferences by assuming a "tolerance" level $\lambda \in (0, 1]$ in the relaxed matching between different function/predicate symbols.

In [8], the formal notion of *fuzzy least Herbrand model* $M_{P,R} : B_L \mapsto [0, 1]$ of the program P with respect to the Similarity R is defined by setting for any $A \in B_L$:

$$\begin{aligned} M_{P,R}(A) &= Sup\{\lambda \in [0,1]\,|A \in M_{H_\lambda(\Gamma(P))}\} \\ &= Sup\{\lambda \in [0,1]\,|H_\lambda(\Gamma(P)) \vDash A\ \} \end{aligned}$$

Roughly speaking, for any $A \in B_L$ the value $M_{P,R}(A)$ provides the best deduction degree of A, i.e, the best level of approximation λ that allows us to prove A by considering an extended program $H_\lambda(\Gamma(P))$. It can be proved that:

$$\begin{aligned} M_{P,R}(A) &= Sup\{\lambda \in [0,1]\,|t_\lambda(A) \in M_{P_\lambda}\} \\ &= Sup\{\lambda \in [0,1]\,|P_\lambda \vDash \tau_\lambda(A)\} \end{aligned}$$

Thus, in order to compute the fuzzy least Herbrand model of a program P extended with a Similarity R, we can equivalently perform our computations in the extended or in the abstract domain.

3 On aggregations in multi context-based Logic Programming framework

In many situations, there can be more than one similarity relation defined in a universe. For example that we have a set of elements defined by some features. We can generate a similarity relation from each feature. In these cases, we must manage and use such information in an appropriated way, for instance we may need to aggregate the obtained relations. In this Section we give a first formal environment in order to make that.

3.1 Aggregation of similarity relations by intersection

The most common way to put together a family of T-transitive fuzzy relations is by calculating their minimum (or infimum), which also is a T-transitive relation. Indeed the following well-known proposition [33] states that x, y are related with respect to R if and only if they are related with respect to all the relations of the family (because the infimum is used to model the universal quantifier $\forall$ in fuzzy logic [9]).

Proposition 4. *Let $(R_i)_{i\in I}$ be a family of T-transitive fuzzy relations on a set U. The relation defined for all $x, y \in U$ by*

$$R(x, y) = min_{i\in I} R_i(x, y)$$

is a T-transitive fuzzy relation on U.

In particular,

Corollary 1. *Let $R_1, \ldots, R_n$ be n similarity relations on a set U. The relation $R_{min} = \bigcap_i R_i$ defined for all $x, y \in U$ by*

$$R_{min}(x, y) = min\{R_1(x, y), \ldots, R_n(x, y)\}$$

is a similarity relation on U.

Example 2. We consider the similarity relations R_1 and R_2 defined in Example 1. Then,

$R_{min}(x, y) = R_{min}(y, x)$

$R_{min}(x, y) = 1 \quad if \ x = y$

$R_{min}(E, H) = .8$

$R_{min}(W, T) = R_{min}(C, D) = .4$

$$R_{min}(G, M) = R_{min}(D, W) = R_{min}(D, T) = R_{min}(C, T) = R_{min}(C, W) = R_{min}(B, T) = \\ = R_{min}(B, C) = R_{min}(B, W) = R_{min}(B, D) = .2$$

$R_{min}(x, y) = 0 \quad otherwise.$

3.2 Aggregation of similarity relations by union

The binary fuzzy relation $R_{min} = \bigcap_i R_i$ is an extreme case of aggregation rule because it is very restrictive. Indeed, $R_{min}(a, b) \leq R_i(a, b)$ for any i. Many times this way to aggregate fuzzy relations by intersection leads to undesirable results in applications. The reason is that the minimum has a drastic effect. For instance, if two objects of our universe are very similar or indistinguishable for all but one similarity relation, and for this particular one the similarity value is very low, then the result applying the minimum will give this last measure and will lose the information of all the other ones. This can be reasonable and useful if we need a perfect matching with respect to all our relations, but this is not the case in many situations. When we need to take all the relations into account in a less drastic way, we need to use other ways to aggregate them. A possibility of softening the previous proposition is by replacing the intersection by the union.

We define $R_{max} = \bigcup_i R_i$ by setting $R_{max}(x, y) = max\{R_1(x, y), \ldots, R_n(x, y)\}$.

Note that R_{max} not is a similarity relation.

Example 3. We consider the similarity relations R_1 and R_2 defined in Example 1. Then,

$R_{max}(x, y) = R_{max}(y, x)$

$R_{max}(x, y) = 1 \quad if \ x = y$

$R_{max}(M, G) = R_{max}(G, W) = R_{max}(T, S) = R_{max}(W, D) = R_{max}(E, H) = .8$

$R_{max}(M, D) = R_{max}(B, S) = R_{max}(B, T) = R_{max}(C, D) = R_{max}(T, D) = .6$

$$R_{max}(C, W) = R_{max}(T, C) = R_{max}(M, E) = R_{max}(M, H) = R_{max}(M, C) = R_{max}(B, G) = \\ = R_{max}(B, W) = R_{max}(G, S) = R_{max}(G, T) = R_{max}(E, C) = R_{max}(E, D) = \\ = R_{max}(H, C) = R_{max}(H, D) = R_{max}(S, W) = R_{max}(T, W) = .4$$

$$R_{max}(M, B) = R_{max}(M, S) = R_{max}(M, T) = R_{max}(W, M) = R_{max}(B, E) = R_{max}(B, H) = \\ = R_{max}(P, E) = R_{max}(B, C) = R_{max}(B, D) = R_{max}(G, E) = R_{max}(G, H) = R_{max}(G, C) = \\ = R_{max}(P, H) = R_{max}(G, D) = R_{max}(E, S) = R_{max}(E, T) = R_{max}(E, W) = R_{max}(H, S) = \\ = R_{max}(H, T) = R_{max}(H, W) = R_{max}(S, C) = R_{max}(S, D) = .2$$

$R_{max}(x, y) = 0 \quad otherwise.$

Note that
$\mathrm{R_{max}(W, D) = 0.8, R_{max}(B, W) = 0.4, R_{max}(B, D) = 0.2}$
which violates the min-transitivity property because
$\mathrm{R_{max}(B, D) \ngeq R_{max}(W, D) \wedge R_{max}(B, W)}$

Let us recall that the max-T product [34, 33] allows us to construct the transitive closure of a given reflexive and symmetric fuzzy relation.

Definition 5. *Let T be a T-norm and R, S two fuzzy relations in a set U. The max-T (or sup-T) product $R \circ S$ of R and S is the fuzzy relation on U defined by*

$$(R \circ S)(x, y) = \sup_{z \in U} T(R(x, z), S(z, y)) \ for \ any \ x, y \in U.$$

Assuming T continuous, due to the associativity of max-T product, we can define for each $n \in N$ the power R^n of a fuzzy relation R recursively:

$R^1 = R$,

$R^{n+1} = R \circ R^n$ for any $n \in N$.

Definition 6. *The T-transitive closure (or T-closure R^* of a fuzzy relation R on a set U is defined by* $R^* = \sup_{n \in N} R^n$.

Proposition 5. *If R is a reflexive and symmetric fuzzy relation on a finite set U of cardinality n, then* $R^* = R^{n-1}$.

Proposition 6. *Let R be a reflexive and symmetric fuzzy relation on a set U.*

$R = R^*$ *if and only if* $T(R(x, y), R(y, z)) \leq R(x, z)$ *for any* $x, y, z \in U$.

Therefore, the transitive closure R^* of a reflexive and symmetric fuzzy relation is a similarity operator. Moreover, it is straightforward to prove that R^* is a relation greater or equal than R ($R^* \geq R$). Moreover,it can be shown [33] that if E is a similarity operator greater or equal than R, then $E \geq R^*$. In other words, the transitive closure R^* of R is the smallest similarity operator that contains R and is therefore the best upper approximation of R.

In fact, the following proposition can be proved.

Proposition 7. *Given a reflexive and symmetric fuzzy relation R on a set U and R^* its transitive closure. Let A be the set of similarity operators on U greater than or equal to R. Then*

$$R^*(x, y) = \inf_{E \in A} \{E(x, y)\}.$$

Example 4. The Min-transitive closure of R_{max} defined in Example 3 is

$R^*_{max}(x, y) = R^*_{max}(y, x)$

$R^*_{max}(x, y) = 1 \quad if \ x = y$

$$R^*_{max}(M, G) = R^*_{max}(M, D) = R^*_{max}(M, W) = R^*_{max}(G, W) = R^*_{max}(G, D) = R^*_{max}(E, H) = \\ = R^*_{max}(T, S) = R^*_{max}(W, D) = .8$$

$$R^*_{max}(M, C) = R^*_{max}(M, B) = R^*_{max}(M, S) = R^*_{max}(M, T) = R^*_{max}(B, S) = R^*_{max}(B, T) = \\ = R^*_{max}(B, G) = R^*_{max}(B, W) = R^*_{max}(B, C) = R^*_{max}(B, D) = R^*_{max}(G, S) = \\ = R^*_{max}(G, T) = R^*_{max}(G, C) = R^*_{max}(C, D) = R^*_{max}(T, D) = R^*_{max}(C, W) = \\ = R^*_{max}(T, C) = R^*_{max}(S, W) = R^*_{max}(T, W) = R^*_{max}(S, C) = R^*_{max}(S, D) = .6$$

$$R^*_{max}(M, E) = R^*_{max}(M, H) = R^*_{max}(B, E) = R^*_{max}(B, H) = R^*_{max}(G, E) = R^*_{max}(G, H) = \\ = R^*_{max}(E, C) = R^*_{max}(E, D) = R^*_{max}(E, S) = R^*_{max}(E, T) = R^*_{max}(E, W) = \\ = R^*_{max}(H, C) = R^*_{max}(H, D) = R^*_{max}(H, S) = R^*_{max}(H, T) = R^*_{max}(H, W) = .4$$

$$R^*_{max}(M, P) = R^*_{max}(B, P) = R^*_{max}(G, P) = R^*_{max}(E, P) = R^*_{max}(H, P) = R^*_{max}(S, P) = \\ = R^*_{max}(T, P) = R^*_{max}(C, P) = R^*_{max}(W, P) = R^*_{max}(D, P) = .2.$$

which is a similarity relation.

3.3 Exploiting R_{min} and R_{max} in the similarity based Logic Programming

The fuzzy relations $R_{min} = \bigcap_i R_i$, and $R_{max} = \bigcup_i R_i$ are two extreme cases of aggregation rules. The first one implies that $R_{min}(a, b) \leq R_i(a, b)$ for any i, the second one that $R_{max}(a, b) \geq R_i(a, b)$ for any i. In the sequel we study properties of these relations in the framework of the Similarity-based Logic Programming. Let us start with R_{min}.

Proposition 8. *Let $R_1, \dots, R_n$ be n similarity relations on a set U, let $\lambda \in (0,1]$ and let P be a logic program. Then,*

$H_{\lambda,R_{\min}}(\Gamma(P)) \subseteq \bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))$

Proof. $A \in H_{\lambda,R_{\min}}(\Gamma(P)) \Longrightarrow \exists A' \in \Gamma(P)$ t.c. $R_{\min}(A, A') \geqslant \lambda$

$\Longrightarrow \min\{R_1(A, A'),\dots,R_n(A, A')\} \geqslant \lambda \Longrightarrow R_i(A, A') \geqslant \lambda\ \forall i = 1, \dots, n$

$\Longrightarrow A \in H_{\lambda,R_i}(A')\ \forall i = 1, \dots, n,\ A' \in \Gamma(P) \Longrightarrow A \in H_{\lambda,R_i}(\Gamma(P))\ \forall i = 1, \dots, n$

$\Longrightarrow A \in \bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))$

Let us prove that the inverse inclusion does not hold.

Let the following logic program be given

P = {q(c)←; r(c)←; p(a)←}

Let us suppose that R_1 and R_2 are two similarity relations defined in $L(P)$ such that

$R_1(r, q) < \lambda$, $R_1(p, q) \geqslant \lambda$, $R_1(r, p) < \lambda$;

$R_2(r, q) < \lambda$, $R_2(p, q) < \lambda$, $R_2(r, p) \geqslant \lambda$.

Then it follows:

$H_{\lambda,R_1}(\Gamma(P))$ = {q(c)←; r(c)←; p(a)←;p(c)←; q(a)←}

$H_{\lambda,R_2}(\Gamma(P))$ = {q(c)←; r(c)←; p(a)←;p(c)←; r(a)←}

$\bigcap_{i=1}^{2} H_{\lambda,R_i}(\Gamma(P))$ = {q(c)←; r(c)←; p(a)←;p(c)←}

$R_{\min}(r, q) < \lambda$, $R_{\min}(p, q) < \lambda$, $R_{\min}(r, p) < \lambda$

$H_{\lambda,R_{\min}}(\Gamma(P))$= {q(c)←; r(c)←; p(a)←}

Results

$\bigcap_{i=1}^{2} H_{\lambda,R_i}(\Gamma(P)) \nsubseteq H_{\lambda,R_{\min}}(\Gamma(P))$. ⋄

By the previous result, because the Logic Programming is a monotonic inference system, it follows that:

$M_{H_{\lambda,R_{\min}}(\Gamma(P))} \subseteq M_{\bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))}$

Therefore, fixed $\lambda \in (0,1]$, the extended program w.r.t. R_{min} allows to deduce less ground atomic formulae than to the intersection of the extended programs of the similarity relations $R_1, \dots, R_n$. Moreover, we can prove the following result:

Proposition 9. *Let $R_1, \dots, R_n$ be n similarity relations on a set U, let $\lambda \in (0,1]$ and let P be a logic program. Then,*

$M_{\bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))} \subseteq \bigcap_{i=1}^{n} M_{H_{\lambda,R_i}(\Gamma(P))}$

Proof. $A \in M_{\bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))} = \bigcup_{j \geqslant 0} T^j_{\bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))}(\emptyset) \Longrightarrow \exists k \geqslant 1$ t.c. $A \in T^k_{\bigcap_{i=1}^{n} H_{\lambda,R_i}(\Gamma(P))}(\emptyset)$

$\Longrightarrow \exists k \geqslant 1$ t.c. $A \in T^k_{H_{\lambda,R_i}(\Gamma(P))}(\emptyset)\ \forall i = 1, \dots, n$

$\Longrightarrow A \in \bigcup_{j \geqslant 0} T^j_{H_{\lambda,R_i}(\Gamma(P))}(\emptyset) = M_{H_{\lambda,R_i}(\Gamma(P))}\ \forall i = 1, \dots, n$

$\Longrightarrow A \in \bigcap_{i=1}^{n} M_{H_{\lambda,R_i}(\Gamma(P))}$

Now let us prove that the inverse inclusion does not hold.

Let the following logic program be given

P = {q(a)←q(b); q(c)←}

M_P = {q(c)}

Let us suppose that R_1 and R_2 are two similarity relations defined in $L(P)$ such that

$R_1(a,b) < \lambda$, $R_1(a,c) \geqslant \lambda$, $R_1(b,c) < \lambda$;

$R_2(a,b) < \lambda$, $R_2(a,c) < \lambda$, $R_2(b,c) \geqslant \lambda$.

Then, it results
$H_{\lambda,R_1}(\Gamma(P)) = \{q(a) \leftarrow q(b);\ q(c) \leftarrow;\ q(c) \leftarrow q(b);\ q(a) \leftarrow\}$
$M_{H_{\lambda,R_1}(\Gamma(P))} = \{q(a),\ q(c)\}$;
$H_{\lambda,R_2}(\Gamma(P)) = \{q(a) \leftarrow q(b);\ q(c) \leftarrow;\ q(a) \leftarrow q(c);\ q(b) \leftarrow\}$
$M_{H_{\lambda,R_2}(\Gamma(P))} = \{q(b),\ q(c),\ q(a)\}$;
$\overset{2}{\underset{i=1}{\cap}} M_{H_{\lambda,R_i}(\Gamma(P))} = \{q(a),\ q(c)\}$;
$\overset{2}{\underset{i=1}{\cap}} H_{\lambda,R_i}(\Gamma(P)) = \{q(a) \leftarrow q(b);\ q(c) \leftarrow\}$
$M_{\overset{2}{\underset{i=1}{\cap}} H_{\lambda,R_i}(\Gamma(P))} = \{q(c)\}$
Then it results
$\overset{2}{\underset{i=1}{\cap}} M_{H_{\lambda,R_i}(\Gamma(P))} \not\subseteq M_{\overset{2}{\underset{i=1}{\cap}} H_{\lambda,R_i}(\Gamma(P))}$. ◇

Summarizing,
$M_{H_{\lambda,R_{\min}}(\Gamma(P))} \subseteq M_{\overset{n}{\underset{i=1}{\cap}} H_{\lambda,R_i}(\Gamma(P))} \subseteq \overset{n}{\underset{i=1}{\cap}} M_{H_{\lambda,R_i}(\Gamma(P))}$

Thus, let us define the *fuzzy least Herbrand model* of P w.r.t. the intersection relation R_{min} as:

$$M_{P,R_{min}}(A) = \sup\{\lambda \in [0,\ 1]\ /\ A \in M_{H_{\lambda,R_{min}}(\Gamma(P))}\}$$

It can be easily proved that:

Proposition 10. *Let $R_1, \ldots, R_n$ be n strict similarity relations on a set U. Then,*
$A \in M_P \iff M_{P,R_{min}}(A) = 1$

Analogous properties hold for R^*_{max}.

Proposition 11. *Let $R_1, \ldots, R_n$ be n similarity relations on a set U, let $\lambda \in (0, 1]$ and let P be a logic program. Then,*
$\overset{n}{\underset{i=1}{\cup}} M_{H_{\lambda,R_i}(\Gamma(P))} \subseteq M_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\Gamma(P))}$

Proof. Let us consider $A \in \overset{n}{\underset{i=1}{\cup}} M_{H_{\lambda,R_i}(\Gamma(P))} \Longrightarrow \exists m \in \{1, \ldots, n\}$ such as $A \in M_{H_{\lambda,R_m}(\Gamma(P))} \Longrightarrow$
$\Longrightarrow A \in \underset{j\geqslant 0}{\cup} T^j_{H_{\lambda,R_m}(\Gamma(P))}(\emptyset) \Longrightarrow \exists k \geqslant 1$ such as $A \in T^k_{H_{\lambda,R_m}(\Gamma(P))}(\emptyset) \Longrightarrow \exists k \geqslant 1$ such as $A \in T^k_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\Gamma(P))}(\emptyset)$
$\Longrightarrow A \in \underset{j\geqslant 0}{\cup} T^j_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\Gamma(P))}(\emptyset) = M_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\Gamma(P))}$
Let us prove that the inverse inclusion does not hold.
Let us consider the following logic program
$P = \{p(a) \leftarrow q(c);\ r(a) \leftarrow\}$
Let us suppose that R_1 and R_2 are two similarity relations defined in $L(P)$ such that
$R_1(r, q) \geqslant \lambda$, $R_1(p, q) < \lambda$, $R_1(r, p) < \lambda$, $R_1(a, c) < \lambda$;
$R_2(r, q) < \lambda$, $R_2(p, q) < \lambda$, $R_2(r, p) < \lambda$, $R_2(a, c) \geqslant \lambda$.
Then it follows:
$H_{\lambda,R_1}(\Gamma(P)) = \{p(a) \leftarrow q(c);\ r(a) \leftarrow;\ q(a) \leftarrow;\ p(a) \leftarrow r(c)\}$
$M_{H_{\lambda,R_1}(\Gamma(P))} = \{q(a),\ r(a)\}$;
$H_{\lambda,R_2}(\Gamma(P)) = \{p(a) \leftarrow q(c);\ r(a) \leftarrow;\ p(c) \leftarrow q(c);\ p(a) \leftarrow q(a);\ r(c) \leftarrow\}$
$M_{H_{\lambda,R_2}(\Gamma(P))} = \{r(c),\ r(a)\}$;
$\overset{n}{\underset{i=1}{\cup}} M_{H_{\lambda,R_i}(\Gamma(P))} = \{r(c),\ r(a),\ q(a)\}$;

$\overset{2}{\underset{i=1}{\cup}}H_{\lambda,R_i}$($\mathit{\Gamma}$(P)) = {p(a)←q(c); r(a)←; q(a)←; p(a)←r(c); p(c)←q(c)←; p(a)←q(a); r(c)←}
H$_{\lambda,R_i}$($\mathit{\Gamma}$(P)) = {q(a)←q(b); q(c)←}
M$_{\overset{2}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\mathit{\Gamma}(P))}$ = {r(c), r(a), q(a), p(a)}
Then, it follows:
M$_{\overset{2}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\mathit{\Gamma}(P))} \nsubseteq \overset{2}{\underset{i=1}{\cup}}M_{H_{\lambda,R_i}(\mathit{\Gamma}(P))}$. ◇

Furthermore, it results that

Proposition 12. *Let $R_1, \dots, R_n$ be n similarity relations on a set U, let $\lambda \in (0,1]$ and let P be a logic program. Then,*
$\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\mathit{\Gamma}(P)) \subseteq H_{\lambda,R^*_{\max}}(\mathit{\Gamma}(P))$

Proof. C ∈ $\overset{n}{\underset{i=1}{\cup}}H_{\lambda,R_i}$($\mathit{\Gamma}$(P)) $\Longrightarrow$ ∃j ∈ {1,. . . , n} t.c. C ∈ H$_{\lambda,R_j}$($\mathit{\Gamma}$(P)) $\Longrightarrow$

$\Longrightarrow$ ∃C' ∈ $\mathit{\Gamma}$(P) t.c. R$_j$(C, C') $\geqslant \lambda \Longrightarrow$ R$_{\max}$(C, C') $\geqslant \lambda \Longrightarrow$ R$^*_{\max}$(C, C') $\geqslant \lambda \Longrightarrow$
$\Longrightarrow$ C ∈ H$_{\lambda,R^*_{\max}}$($\mathit{\Gamma}$(P))
Let us prove that the inverse inclusion does not hold.
Let us consider two similarity relations R$_1$ ed R$_2$ defined in a first order languages L such that
R$_1$(a, b) = 0.3, R$_1$(a, c) = 0.3, R$_1$(b, c) = 0.5
R$_2$(a, b) = 0.6, R$_2$(a, c) = 0.4, R$_2$(b, c) = 0.4
then
R$_{\max}$(a, b) = 0.6, R$_{\max}$(a, c) = 0.4, R$_{\max}$(b, c) = 0.5
and
R$^*_{\max}$(a, b) = 0.6, R$^*_{\max}$(a, c) = 0.5, R$^*_{\max}$(b, c) = 0.5
Let P = {a←}
Then, results that
H$_{0.5,R_1}$($\mathit{\Gamma}$(P)) = {a←}
H$_{0.5,R_2}$($\mathit{\Gamma}$(P)) = {a←; b←}
$\overset{2}{\underset{i=1}{\cup}}H_{0.5,R_i}$($\mathit{\Gamma}$(P)) = {a←; b←}
H$_{0.5,R^*_{\max}}$($\mathit{\Gamma}$(P)) = {a←; b←; c←}
Then,
H$_{0.5,R^*_{\max}}$($\mathit{\Gamma}$(P)) $\nsubseteq \overset{2}{\underset{i=1}{\cup}}H_{0.5,R_i}$($\mathit{\Gamma}$(P)) ◇

Then, because the Logic Programming is a monotonic inference system
M$_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\mathit{\Gamma}(P))} \subseteq$ M$_{H_{\lambda,R^*_{\max}}(\mathit{\Gamma}(P))}$
Therefore, fixed $\lambda \in (0,1]$, the extended program w.r.t. R^*_{max} allows to deduce more ground atomic formulae than to the union of the extended programs of the similarity relations $R_1, \dots, R_n$.

Summarizing,
$\overset{n}{\underset{i=1}{\cup}}M_{H_{\lambda,R_i}(\mathit{\Gamma}(P))} \subseteq$ M$_{\overset{n}{\underset{i=1}{\cup}} H_{\lambda,R_i}(\mathit{\Gamma}(P))} \subseteq$ M$_{H_{\lambda,R^*_{\max}}(\mathit{\Gamma}(P))}$

Let us define the *fuzzy least Herbrand model* of P w.r.t. the union relation R^*_{max} as:

$$\text{M}_{P,R^*_{max}}(\text{A}) = \sup\{\lambda \in [0,\ 1]\ /\ \text{A} \in \text{M}_{H_{\lambda,R^*_{max}}(\mathit{\Gamma}(P))}\}$$

It can be proved that:

Proposition 13. *Let $R_1, \ldots, R_n$ be n strict similarity relations on a set U and let P be a logic program. Then,*
$A \in M_P \iff M_{P,R^*_{max}}(A) = 1$

4 Conclusion

In this paper we studied two estreme cases (R_{min} and R_{max}) of group preferences relationsimposing the min-transitivity property. The operators Min and Max (intersection and union in terms of fuzzy relations) are two extreme cases of aggregation rules. Properties of these relations in the framework of Similarity-based Logic Programming have been proved. As future work different aggregation rules will be studied in order to manage the similarity values in a less extreme way.

Acknowledgements
The authors gratefully acknowledge the referee for their useful comments which have contributed to improve this work.

References

1. Apt R.K., *Logic Programming*, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, vol. B, (Elsevier, Amsterdam, 1990) 492-574.
2. Bezdek JC, Harris JO. *Fuzzy partitions and relations: An axiomatic basis for clustering.* Fuzzy Sets Syst 1978;1:112127.
3. Bodenhofer U. *A Similarity-Based Generalization of Fuzzy Orderings.* (1999) Vol. C 26 of Schriftenreihe der Johannes-Kepler-Universitat Linz, Universitatsverlag Rudolf Trauner.
4. Dubois D., Prade H., *Resolution principles in possibilistic logic*, Int. Journal of Approximate Reasoning, 3 (1990) 1-21.
5. J.C. Fodor and S. Ovchinnikov, *On aggregation of T-transitive fuzzy binary relations.* Fuzzy Sets and Systems 72 (1995) 135-145. (MR 1335529)
6. Fodor, J., Roubens, M. *Fuzzy preference modelling and multicriteria decision support.* Kluwer Ac. Publ., 250 pages, ISBN 0-7923-3116-8, 1994.
7. Genesereth M.R., Ketchpel S.P. *Software agents.* Communications of the ACM, 37(7):48-53, 1994.
8. Gerla G., Sessa M.I., *Similarity in Logic Programming*, in: G. Chen, M. Ying, K.-Y. Cai (Ed.s), Fuzzy Logic and Soft Computing, (Kluwer Acc. Pub., Norwell, 1999), 19-31.
9. Hajek P. *Metamathematics of fuzzy logic.* Dordrecht, The Netherlands: Kluwer; 1998.
10. Ishizuka M., Kanai N., *PROLOG-Elf incorporating fuzzy logic*, Proc. 9th Int. Joint. Conf. on Artificial Intelligence (Springer, Berlin, 1985) 701-703.
11. Joan Jacas, Jordi Recasens: Aggregation of T-transitive relations. Int. J. Intell. Syst. 18(12): 1193-1214 (2003)
12. Kifer M., Li A., *On the Semantic of Rule-Baaed Expert Systems with Uncertainty*, Inter. Conf. on Databases Theory 1988, Lecture Notes in Computer Science, vol. S26 (Springer, Berlin, 1988) 186-202.
13. Kifer M., Subrahmanian V.S., *Theory of Generalized Annotated Logic Programming and its Applications*, Journal of Logic, Programming 12(3&4) (1992) 335-367.
14. Kowalski, R.A., *Predicate logic as a programming language*, in: Proc. IFIP'74 (1974) 569-574.
15. Formato F., Gerla G., Sessa M.I., *Similarity-based unification*, Fundamenta Informaticae 40 (2000) 1-22.
16. Loia V., Luongo P., Senatore S. and Sessa M.I., *A Similarity-based View to Distributed Information Retrieval with Mobile Agents.* In The 10th IEInternational Conference on Fuzzy Systems, Melbourne, Australia, December 2-5, 2001.

17. Loia V., Senatore S. and Sessa M.I., *Similarity-based agents for email mining.* In Proc. of the joint Conference IFSA/NAFIPS 2001, Vancouver, Canada, July 25-28, 2001.
18. Loia V., Senatore S. and Sessa M.I., *Similarity-based SLD Resolution and its implementation in an Extended Prolog System*, Proceedings of 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, December 2-5 2001. IEEE PRESS.
19. Martin T.P., Baldwin J.F., Pilsworth B.W., *The implementation of FPROLOG a fuzzy PROLOG interpreter*, Fuzzy Sets and Systems 23 (1987) 119-129.
20. Mukaidono M., Shen Z.L., Ding L., *Fundamentals of fuzzy PROLOG*, Int. Journal of Approximate Reasoning 3 (1989) 179-193.
21. Ovchinnikov S.V. *Aggregating transitive fuzzy binary relations.* IJUFKBS 1995;3:4755.
22. Ovchinnikov S.V. *Similarity relations, fuzzy partitions, and fuzzy orderings.* Fuzzy Sets and Systems, (1991) 40, 107-126.
23. Ovchinnikov S.V. *Transitive fuzzy orderings of fuzzy numbers, Fuzzy Sets and Systems.* V.30 n.3, p.283-295, May 10, 1989
24. Ovchinnikov, S. V. *Representations of transitive fuzzy relations.* In: Skala, Termini, and Trillas, 1983;105-118.
25. Poincaré J.H. *La Valeur de la Science.* Flammarion, Paris (1905).
26. Pradera A, Trillas E, Castineira E. *On the aggregation of some classes of fuzzy relations.* In: Bouchon-Meunier B, Gutierrez J, Magdalena L, Yager R, editors. Technologies for constructing intelligent systems. Heidelberg: Springer-Verlag; 2002. pp 125147.
27. Pradera A, Trillas E. *A note on pseudometrics aggregation.* Int J Gen Syst. 2002;31(1):4151.
28. Saminger S., Mesiar R., Bodenhofer U. *Domination of Aggregation Operators and Preservation of Transitivity.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(Supplement): 11-36 (2002)
29. Sessa M.I., *Approximate Reasoning by Similarity-based SLD Resolution*, Theoretical Computer Science, 275 (2002) 389-426.
30. Sessa M.I. *Translations and Similarity-based Logic Programming*, Soft Computing 5(2) (2001).
31. Subrahmanian V.S. *On the semantics of quantitative logic*, Proc. IEEE Symposium on Logic Programming (1987) 173-182.
32. Trillas E., Valverde L. *An Inquiry into Indistinguishability Operators.* In: H.J. Skala, S. Termini, E. Trillas (eds.), Aspects of Vagueness. Reidel, Dordrecht (1984), 231-256.
33. Valverde L. *On the structure of F-indistinguishability operators.* Fuzzy Sets Syst 1985;17:313328.
34. Zadeh, L.A., *Similarity Relations and Fuzzy Orderings.* Information Sciences 3 (1971) 177-200.