A fuzzy EL description logic with crisp roles and fuzzy aggregation for web consulting

Peter Vojtáš
Charles University and ICS AS, Prague
Peter.Vojtas@mff.cuni.cz

Abstract

In semantic web a query can contain several vague concepts of user's gradual preferences. This particular preferences need to be combined to get an overall ordering of results. We propose a fuzzy description logic with existential restrictions, crisp roles, fuzzy concepts and fuzzy combining functions (fuzzy aggregation operators). For web consulting we need to solve the instance problem. We show some results on polynomial complexity of instance problem.

Keywords: fuzzy description logic with existential restrictions, fuzzy concept, user preference query, fuzzy aggregation operator, instance problem

1 Introduction and motivation

In the semantic web context, information has to be retrieved, processed, shared, reused and aligned in an automatic way by software agents.

In [16] G. Stoilos et al. describe experience with applications in the semantic web. They have shown that these applications are rarely a matter of true or false answers, but rather procedures that require degree of relatedness, similarity or ranking. Similar motivation led to development of fuzzy description logic for the semantic web in [18] and [17] by U. Straccia. See also specialized workshops and proceedings [19] and [7].

Exact constraints of a query often lead to empty or too many answers. Using fuzzy atomic concepts, we can better express gradualism of user preferences. Another source of fuzziness referred in the literature are uncertain or vague values.

In fuzzy databases [6] J. Galindo et al. describe fuzzy attribute type 1 - "these are represented as usual attributes because they do not allow fuzzy values. Nevertheless, information is stored in the fuzzy background knowledge base about the nature or context of them. They are classical attributes that admit fuzzy processing." By our opinion, this coincides with information on the web. Information on the web can be vague or imprecise, but there is no degree of fuzziness attached to it (web is "read only"). Source of fuzziness is in vagueness and preferences of user's query and his/her interpretation of data.

Motivation example. Imagine a user U looking for a hotel which is *cheap*, *close to a beach*, and has a *new building*. Let us first assume our data are as in table 1 (where distance is the distance to the nearest beach and yoc is the year of construction).

Table 1: Hotel attribute values .

Hotel	**price**	**distance**	**yoc**
h1	150	300	1980
h2	200	450	2000

Particular attribute preferences of the user U can be expressed by fuzzy subsets

$f^U_{cheap}, f^U_{close}, f^U_{new}$ of particular attribute domains , e.g. $f^U_{cheap}(150) = 0.75$, $f^U_{close}(300) = 0.6$, $f^U_{new}(1980) = 0.2$ and similarly for other values.

This results in a table with user's U degree of preferences of particular attribute in table 2.

Table 2: U-Degree of hotel attribute score

Hotel	**cheap_U**	**close_U**	**new_U**
h1	0.75	0.6	0.2
h2	0.5	0.3	0.9

Notice that no hotel is clearly better in all attributes.

Here the main point of our motivation comes. Practical experiences have shown that comparison of user's overall ordering of objects with score of particular attributes is seldom a conjunctive or disjunctive combination (see e.g. [20]). In databases it means that some orderings cannot be described by neither conjunctive nor disjunctive queries. We need a more general combination (fusion, aggregation) of different features of a query. One solution is to work with a fuzzy aggregation (e.g. weighted sum), which can order objects with incomparable particular attributes. By an inductive method ([20]) we can learn the combination function of user U to be

$$@_U(cheap\_U, close\_U, new\_U) =$$

$$= \frac{2 * cheap\_U + 3 * close\_U + new\_U}{6}$$

.

This for hotel h1 gives

$$@_U(0.75, 0.6, 0.2) =$$

$$= \frac{2 * 0.75 + 3 * 0.6 + 0.2}{6} = \frac{3.5}{6} = 0.58...$$

and this is an overall degree with which the hotel h1 is good for the user U. But not the numerical value counts here, it is rather the comparison of overall degree for both hotels. For the hotel h2 we get $@_U(0.5, 0.3, 0.9) = 0.46$, so, global score of hotel h2 is strictly smaller, hence the hotel is less preferable for user U than h1 (of course, for another user with a different combination function this need not be the case).

This feature of querying, namely combination of particular attribute score to a global score, was already studied in GAP - generalized annotated programs of M. Kifer and V. S. Subrahmanian [11], information retrieval by R. Fagin [4], database rank aware querying by I. F. Ilyas et al in [9] and D. Papadias et al [15], just to mention a few.

Example on the web.

We consider a model of web services presented by D. Fensel et al. in [5] (depicted bellow, boxed items denote data and circled items denote processes).

$$\begin{array}{ccccc}
\boxed{G} & \longrightarrow & WSD & \longrightarrow & \boxed{WS} \\
\uparrow & & & & \downarrow \\
GD & & & & SD \\
\uparrow & & & & \downarrow \\
\boxed{UD} & & & & \boxed{AS}
\end{array}$$

Main point of the [5] model is the distinction between user desire UD and abstracted goal G processable by the system. Second distinction is between a web service WS, understood as a computational entity accessible over the Internet, and an actual service AS (discovered via this). In this setting [5] recognize goal discovery GD - a process abstracting from individual and specific features of the user query to a semantically annotated query. Web service discovery WSD is based on matching abstracted goal description with semantic annotation of web services. This is related to our problem, if we assume that semantic is expressed by description logic or a web modeling language based on description logic. Service discovery SD is a processes recognizing an actual service from a set of services available at WS.

In this paper we describe our task in this realm of web services.

The paper is organized as follows: first we describe the syntax and semantics of a fuzzy description logic $f-\mathcal{EL}^{\mathcal{G}}_{cr}$. Further we discuss the instance problem for $f-\mathcal{EL}^{\mathcal{G}}_{cr}$ from the point of view of web consulting. We conclude with some observations, comparison and plans for future research.

2 A fuzzy description logic $f-\mathcal{EL}^{\mathcal{G}}_{cr}$

In order agents and services can negotiate and find an answer, we have to consider our problem as a part of web modeling languages. As description logic has become a part of standards for the semantic web, we would like to have these features in models of description logic. A standard reference for description logic is F. Baader et al. [2].

In this paper we propose a fuzzy $\mathcal{EL}$-type description logic (for references on $\mathcal{EL}$ description logic see R. Kuesters and R. Molitor [13]). We look for a logic which has minimal necessary tools to allow construction of concepts which aggregate particular attribute score to global score according to user preferences, as in our example.

In this section we introduce a description logic which in some parameters (e.g. crisp roles, without negations) is a weakening of fuzzy description logic of U. Straccia [17], [18] and in some parameters is a strengthening (aggregations). Moreover we use only existential restrictions which have surprisingly great expressive power wrt. applications ([2], [13]). We loose the ability to describe fuzziness in roles (e.g. uncertainty in values) but we gain combination of particular user preferences to a global score. Intended meaning is, complex concepts describe user query and atomic concepts play the role of selection conditions (similarly as in WHERE expressions in an SQL query).

Note that we do not have negation in our logic because it can be hidden in atomic concepts (similarly as SQL selection conditions are usually closed on negation).

Our alphabet consists of (mutually disjoint) sets N_C of concepts names containing $\top$, N_R role names, N_I instance names.

Our example in description logic. Basic building boxes of description logic are concepts and roles. Here, roles express data (properties of resources - hotels in our case) stored in crisp data. Although the basic model of expressing triplets "resources - properties - values" are oriented graphs, we use here the language of logic.

We describe an example of the $f-\mathcal{EL}^{\mathcal{G}}_{cr}$ description logic alphabet (subscript cr stands here for "crisp roles"). Our $f-\mathcal{EL}^{\mathcal{G}}_{cr}$ alphabet $\mathcal{L}^{ex}$ consists of roles names

$N^{ex}_R = \{hotel\_price, hotel\_distance, hotel\_yoc\}$.

atomic concept names

$$N^{ex}_C = \{cheap\_U, close\_U, new\_U, \\ cheap\_hotel\_U, close\_beach\_hotel\_U, \\ new\_hotel\_U, good\_hotel\_U\}.$$

Some of them we sometimes abbreviate as $Hcheap\_U, Hclose\_U, Hnew\_U$.

Further, our language $\mathcal{L}^{ex}$ contains instance names (either typed or untyped), this can look as follows

$N^{ex}_I = \{h1, h2, 150, 300, 1980, \ldots\}$.

Our language of description logic further contains constructor $\exists$ and a finite set $\mathcal{G}$ of combination functions symbols ($\mathcal{G}$ standing for **g**lobal scoring functions) and the arity function $ar : \mathcal{G} \longrightarrow \{n \in N : n \geq 2\}$.

In our case $\mathcal{G}^{ex} = \{@_U\}$.

Concept descriptions in $f-\mathcal{EL}^{\mathcal{G}}_{cr}$ are formed according to the following syntax rules

$$C \longrightarrow \top | A | @(C_1, \ldots, C_n) | \exists r.C$$

Example in description logic (continued).

Using above, in our DL $f-\mathcal{EL}^{\mathcal{G}}_{cr}$ we can have more complex concepts like

∃ hotel_price.cheap_U
∃ hotel_distance.close_U
∃ hotel_yoc.new_U

$@_U(Hcheap\_U, Hclose\_U, Hnew\_U)$

Now we have the language $\mathcal{L}^{ex} = \{N_R^{ex}, N_C^{ex}, N_I^{ex}, \exists, \mathcal{G}^{ex}\}$. In order to give this syntax a meaning we have to define interpretations of our language. As already motivated by our example, we are modeling gradual user preferences. The modeling tool for this is many valued logic (which so far did not touch syntax of our description logic).

In our description logic $f - \mathcal{EL}_{cr}^{\mathcal{G}}$ we have interpretations parameterized by a (possibly partially, usually linearly) ordered set of truth values with aggregations.

Interpretations are based on a choice of preference structure. Let $\mathcal{P} = \{P, \leq, \{@^\bullet : @ \in \mathcal{G}\}\}$ be such that $(P, \leq, 1_P, 0_P)$ is a complete lattice and $@^\bullet : P^{ar(@)} \longrightarrow P$ are totally continuous functions in the sense of lattice theory (see [11], this notion requires also functions to be order preserving - so e.g. negation is not totally continuous in the sense of lattice theory). Moreover we assume our global scoring functions fulfill

$$@^\bullet(1_P, \ldots, 1_P) = 1_P$$

and

$$@^\bullet(0_P, \ldots, 0_P) = 0_P$$

hence $@^\bullet$ are fuzzy aggregation operators in the sense of T. Calvo et al. [3]. $\mathcal{P}$ is a preference structure.

Notice, we do not assume our fuzzy $\mathcal{EL}$-type logic contains $\sqcap$ connected to a specific t-norm. Fuzzy aggregations generalize both fuzzy conjunctions and disjunctions. In the case $\mathcal{G} = \{@\}$ is a one element set, we will write $f - \mathcal{EL}_{cr}^{@}$.

For a preference structure $\mathcal{P}$, a $\mathcal{P}$-interpretations is a pair $\mathcal{I} = \langle \Delta^\mathcal{I}, \bullet^\mathcal{I} \rangle$, with nonempty domain $\Delta^\mathcal{I}$ and interpretation of language elements

$a^\mathcal{I} \in \Delta^\mathcal{I}$, for $a \in N_I$ (with unique name assumption)

$A^\mathcal{I} : \Delta^\mathcal{I} \longrightarrow P$, for $A \in N_C$

$r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, for $r \in N_R$.

The $\mathcal{P}$-interpretation $\mathcal{I}$ extends to arbitrary $f - \mathcal{EL}_{cr}^{\mathcal{G}}$ concepts by

$(\exists r.C)^\mathcal{I}(x) = \inf\{C^\mathcal{I}(y) : (x, y) \in r^\mathcal{I}\}$

and

$(@(C_1, \ldots, C_n))^\mathcal{I}(x) = @^\bullet(C_1^\mathcal{I}(x), \ldots, C_n^\mathcal{I}(x))$

Note that interpretation of existential restrictions is a special case of the fuzzy description logic by [17], assuming fuzzy conjunction is a t-norm $\otimes$ fulfilling $\otimes(1_P, p) = p$.

Example of fuzzy interpretations of $f - \mathcal{EL}_{cr}^{\mathcal{G}}$. In our case $P = [0, 1]$ and in a Herbrand like interpretation $\mathcal{H}$ we can have

$h1^\mathcal{H} = h1, 150^\mathcal{H} = 150, \ldots$ and hence
$\Delta^\mathcal{H} = \{h1, h2, 150, 300, 1980, \ldots\}$,
$hotel\_price^\mathcal{H}(h1, 1000)$
$cheap\_U^\mathcal{H}(1000) = 0.75$
$good\_hotel\_U^\mathcal{H}(h1) = .58$

Note, these values coincide with our introductory example but need not be same in all interpretations of our language. We need some analogy of domain specific axioms as it is usual logical systems.

TBox axioms are same as in general description logic of [2] and consist of statements of the form $C \equiv D$ and $C \sqsubseteq D$.

A TBox $\mathcal{T}$ is a finite set of TBox axioms. An interpretation $\mathcal{I}$ is a model of above TBox axioms if $C^\mathcal{I} = D^\mathcal{I}$ and $C^\mathcal{I} \leq D^\mathcal{I}$ (see [17]).

Following U. Straccia [17] we have to define ABox expressions using thresholds. For a $p \in P$, $a \in N_I$ and C an $f - \mathcal{EL}_{cr}^{\mathcal{G}}$ concept, $\langle a : C \geq p \rangle$ is a $\mathcal{P}$-ABox expression. Note, DL syntax and TBoxes do not depend on preference structure, ABox depends on a chosen preference structure $\mathcal{P}$.

A $\mathcal{P}$-ABox is a finite set $\mathcal{A}$ of $\mathcal{P}$-ABox expressions. A fuzzy $\mathcal{P}$-interpretation $\mathcal{I}$ is a model of $\mathcal{A}$ if it satisfies all assertions, especially if $C^\mathcal{I}(a^\mathcal{I}) \geq p$.

Example TBox and ABox axioms.

cheap_hotel_U $\equiv \exists$ hotel_price.cheap_U
Hclose_U $\equiv \exists$ hotel_distance.close_U
new_hotel_U $\equiv \exists$ hotel_yoc.new_U
good_hotel_U $\equiv @_U$(cheap_hotel_U,
close_beach_hotel_U,
new_hotel_U)

and an ABox expression (we do not use concrete domains here) e.g.

$h1 : cheap\_hotel\_U \geq 0.75$

All problems and questions of classical description logic which end with yes-no answer are in fuzzy logic subject to answers with a certain degree - in our case from P. We can formulate a yes-no problem with a threshold (e.g. strong version with true with degree 1 or degree at least some $p \in P$) or as a v-problem (variable-problem) to find best degree true in all models (see e.g. [17], [18] or analogy in fuzzy logic programming [21]).

An equivalence problem $C \equiv_{\mathcal{T},\mathcal{A}} D$ wrt a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ asks whether in all interpretations $\mathcal{I}$ which are model of $\mathcal{T}$ and $\mathcal{A}$ is $C^{\mathcal{I}} = D^{\mathcal{I}}$.

Similarly, a subsumption problem $C \sqsubseteq_{\mathcal{T},\mathcal{A}} D$ questions $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for all $x \in \Delta^{\mathcal{I}}$. Here in the $\leq$ is hidden the question whether the truth value of a many valued implication $C^{\mathcal{I}}(x) \longrightarrow D^{\mathcal{I}}(x)$ equals 1_P for all $x \in \Delta^{\mathcal{I}}$.

We have two version of consistency (satisfiability) problem. A $f - \mathcal{EL}^{\mathcal{G}}_{cr}$ concept C is strongly (weakly resp.) satisfiable wrt $\mathcal{T}$ and $\mathcal{A}$ if there is a $\mathcal{P}$-interpretation $\mathcal{I}$, which is a model of both $\mathcal{T}$ and $\mathcal{A}$ such that there is an $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) = 1_P$ ($C^{\mathcal{I}}(x) > 0_P$ resp.)

Instance problem. An individual a is an instance (for $p \in \mathcal{P}$ a p-instance) of C with respect to a $\mathcal{P}$-ABox $\mathcal{A}$ (denoted $a \in_{\mathcal{A}} C$ $(\geq p)$) if for all $\mathcal{P}$-interpretations $\mathcal{I}$ which are model of $\mathcal{A}$ we have $C^{\mathcal{I}}(a^{\mathcal{I}}) = 1_P$, or $\geq p$ resp. A $p \in \mathcal{P}$ is a correct answer to a v-instance problem $?{-}a : C$ wrt $\mathcal{A}$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq p$ in all models of $\mathcal{A}$ and p is the greatest such element of P.

3 Instance problem in $f - \mathcal{EL}^{\mathcal{G}}_{cr}$ for web consulting

Web consulting example. We continue in our example. In the web service model of D. Fensel et al. [5] we start with a desire of a user U

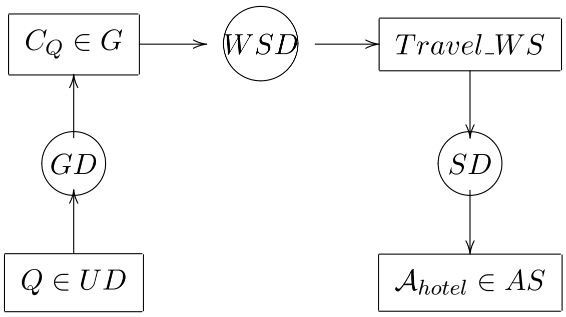
$Q =$

"a hotel which is *cheap*, *close to a beach*, and has a *new building*"

in UD.

The goal discovery process finds an $f - \mathcal{EL}^{\mathcal{G}}_{cr}$ concept $C_Q = good\_hotel\_U$ in the language $\mathcal{L}^{ex}$ (it is not our task to discuss this GD process).

Now, the web service discovery process has to find a service able to answer this query. Assume first, the web service semantic is described in a description logic language with same atomic concepts and roles and allocates a web service $Travel\_WS$ and within it a special service described by an ABox $\mathcal{A}_{hotel}$, as depicted bellow:



Now the problem whether a hotel (e.g. $h1$) fits and in which degree user U requirements based on $\mathcal{A}_{hotel}$ is exactly an instance problem, of $h1 \in_{\mathcal{A}_{hotel}} C_Q$.

Our starting point is a solution of instance problem in crisp description logic $\mathcal{EL}$.

The instance problem for description logic with existential restrictions was shown to be polynomial time solvable by R. Kuesters and R. Molitor in [13] (for acyclic forms). Main idea of their solution is following. Query concepts are represented as (labeled) $\mathcal{EL}$-description trees. ABox is represented as an (labeled) $\mathcal{EL}$-description graph. The instance problem is shown to be equivalent to finding a homomorphic embedding of the tree into the graph (preserving some monotonicity conditions on labels).

This monotonicity condition on labels is a recursive call of a simple subsumption problem along the tree embedding, namely an individual a is in a concept C on the query tree side if there is on the ABox side a concept D containing a and moreover if we know that $D \sqsubseteq_{\mathcal{A}} C$. Then from $a \in_{\mathcal{A}} D$ follows $a \in_{\mathcal{A}} C$.

These monotonicity conditions use a knowledge true in all models (a sort of logical axioms) about the interpretation of $\sqcap$ or an information from a TBox.

In [13] the following tautology is used, valid in all models.

Namely (after a suitable permutation, using commutativity) for all $\mathcal{EL}$ concepts $C_1, \ldots, C_n, C_{n+1}, \ldots, C_{n+m}$ and two valued $\mathcal{EL}$ interpretations $\mathcal{J}$, we have $C_1^{\mathcal{J}} \sqcap \ldots \sqcap C_n^{\mathcal{J}} \sqcap C_{n+1}^{\mathcal{J}} \sqcap \ldots \sqcap C_{n+m}^{\mathcal{J}} \subseteq C_1^{\mathcal{J}} \sqcap \ldots \sqcap C_n^{\mathcal{J}}$ and hence if the query concept requires an individual to be in $\sqcap_{i=1}^{n} C_i$ and in the ABox there is an information that this individual is in $\sqcap_{i=1}^{n+m} C_i$, the requirement is fulfilled. Hence the embedding of the tree into graph can be easily constructed checking inclusion of finite sets of labels $\{C_1, \ldots, C_n, C_{n+1}, \ldots, C_{n+m}\}$.

So, Kuesters-Molitor KM-algorithm is correct, because uses correct inclusions between intersections. The KM-algorithm is also complete, because using only intersection (without negation and union) the only remaining tautologies are equalities of the form $C \cap C \equiv C$, and this is handled by the fact that labels of $\mathcal{EL}$-graphs are sets of concepts appearing in expressions.

So Kuesters-Molitor result can be reformulated as follows.

Theorem [13]. For a preference structure $\mathcal{P}_c = \{\{0,1\}, \leq, 1, 0\}$ the instance problem for $\mathcal{P}_c - \mathcal{EL}_{cr}^{\sqcap}$ is polynomialy solvable.

Notice, that using the inclusion $\cup_{i=1}^{n} C_i \subseteq \cup_{i=1}^{n+m} C_i$ we can get analogous results for $\sqcup$, namely the following holds

Theorem. The instance problem for $\mathcal{P}_c - \mathcal{EL}_{cr}^{\sqcup}$ is solvable in polynomial time.

Just notice, that the $a \in \sqcup_{i=1}^{n} C_i$ has to be fulfilled in the ABox $\mathcal{A}$ and $a \in \sqcup_{i=1}^{n+m} C_i$ should be a part of query concept C_Q tree.

A little bit further, we can exploit also the inclusion $\sqcap_{i=1}^{n} C_i \sqsubseteq \sqcup_{i=1}^{n+m} C_i$, this mixes intersection on the ABox side and union on the query side. Nevertheless, one can imagine that an ABox contains such information and a query such a requirement.

Example of mixing intersection and union between G and WS. In the case of union, it suffices WSD mention less attributes than the query in a disjunctive way. Along [5] a WS can advertise offering its services in Travel and special services either on prices or distance to beach or age of building.

Nevertheless, we have to be careful in formulating a result, namely in [1] using NP-completeness of Monotone 3SAT problem, it is shown that the instance problem for $\mathcal{P}_c - \mathcal{EL}_{cr}^{\{\sqcup,\sqcap\}}$ is co-NP complete, hence practically intractable. Our results reads as follows.

Theorem. Assume $\mathcal{L}_1$ is a $\mathcal{P}_c - \mathcal{EL}_{cr}^{\sqcup}$ and $\mathcal{L}_2$ is a $\mathcal{P}_c - \mathcal{EL}_{cr}^{\sqcap}$ language, and C is a concept either in $\mathcal{L}_1$ or $\mathcal{L}_2$ and $\mathcal{A}$ is an ABox either in $\mathcal{L}_1$ or $\mathcal{L}_2$. Then the instance problem $a \in_{\mathcal{A}} C$ is in PTIME.

In fuzzy case there is an easy and difficult generalization of these results. The easy uses t-norms and conorms, the difficult deals with aggregators.

Assume our preference structure is the standard fuzzy one, namely $\mathcal{P}_f = \{[0,1], \leq, 0, 1\}$ and $\otimes$ is a t-norm and $\oplus$ corresponding dual t-conorm. Moreover assume that $\sqcap_{\otimes}$ and $\sqcup_{\oplus}$ are corresponding description logic connectives. Then we have the following:

Theorem. Assume $\mathcal{L}_1$ is a $\mathcal{P}_f - \mathcal{EL}_{cr}^{\sqcup\oplus}$ and $\mathcal{L}_2$ is a $\mathcal{P}_f - \mathcal{EL}_{cr}^{\sqcap\otimes}$ language, and C is a concept either in $\mathcal{L}_1$ or $\mathcal{L}_2$ and $\mathcal{A}$ is an ABox either in $\mathcal{L}_1$ or $\mathcal{L}_2$. Then there is a correct algorithm for the instance problem $a \in_{\mathcal{A}} C$ in PTIME.

Proof. Note that for a t-norm and dual t-conorm the following inequalities are valid.

$$\otimes_{i=1}^{n+m} C_i \leq \otimes_{i=1}^{n} C_i \leq \oplus_{i=1}^{k} C_i \leq \oplus_{i=1}^{k+l} C_i$$

Completeness of our algorithm depends on the fact, whether these are only connections between $\otimes$ and $\oplus$.

Moreover note, that this is true especially because the truth value computations can be run parallel to KM-algorithm in a bookkeeping procedure along the classical tree embedding (see [21]).

From a application point of view it is hardly to assume that all combination arose from a single t-norm. Even for two t-norms the associativity and commutativity cannot be guaranteed in general. One can study pairs of different t-norms and/or conorms, fulfilling one of (here we relax our notation)

Our approach has an extra feature. Namely fuzzy aggregation (annotations) are a generalization of both conjuctions and disjunctions. The above problem can reformulated and strengthened to the following

Problem. Assume $@_1$ and $@_2$ are two fuzzy annotation operators on $[0,1]$. Characterize those clases of operators $@(C_1, \ldots, C_n, C_{n+1}, \ldots, C_{n+m})$ for which the following is decidable in polynomial time

$(@_1)_{i=1}^{n+m} C_i \leq (@_1)_{i=1}^{n} C_i$ - we call these order conjunctive operators

$(@_1)_{i=1}^{n} C_i \leq (@_1)_{i=1}^{n+m} C_i$ - we call these order disjunctive operators

$@_1(C_1, \ldots, C_n) \leq @_2(D_1, \ldots, D_k)$

There are some initial results in [3].

Practical experiments with the fuzzy inductive logic programming FILP in [20] have shown that (at least for tested data and users) some aggregation operators behave order conjunctive if all values are small and order disjunctive if all values are large. FILP gets aggregations after a discretisation step, and so we can easily prove

Theorem. Assume $@_1$ and $@_2$ are two fuzzy aggregation operators obtained by the FILP method for different vectors of values $\bar{s}$ and $\bar{t}$. Then deciding $@_1(\bar{s}) \leq @_2(\bar{t})$ is polynomialy hard for a class of FILP induced operator.

4 Conclusions

In this paper we have presented a fuzzy description logic $f - \mathcal{EL}_{cr}^{\mathcal{G}}$ (here f refers to fuzzy with possible specification of a preference structure, $\mathcal{G}$ refers to a set of **g**lobal score combination function replacing classical connectives, *cr* refers to crisp roles and $\mathcal{EL}$ refers to description logic using only existential restriction (which in crisp case automatically assumes conjunction, but not in our case)). We gave an extensive motivation for our approach from web consulting domain, which is related to instance problem of $f - \mathcal{EL}_{cr}^{\mathcal{G}}$. We have shown some results on polynomial time complexity of instance problem.

To conclude, we mention that our description logic $f - \mathcal{EL}_{cr}^{\mathcal{G}}$ has semantics equivalent to a variant of generalized annotated programs of [11], for details see [12].

Despite successful standardization efforts by the W3C, there are still numerous different ontology representation languages being used and for practical applications we even need these. P. Hitzler et al. in [8] argue for an OWL subset known as DLP-Description logic programs to be used in applications. Using our results in [21] we can generalize DLP to many valued logic with fuzzy aggregation operators with semantics equivalent to $f - \mathcal{EL}_{cr}^{\mathcal{G}}$.

The realm of instance problem with aggregations changes dramatically if we allow noncyclic constructions. In this case we can even get undecidable problem (using the [11] result on non-continuity of the production operator for restricted semantics of GAP programs, see [12]).

In future we would like to apply these results on projects from network security (see [10]) and job market system (see [14]).

Acknowledgements

Partially supported by Czech projects MSM 0021620838, 1ET100300419 and 1ET100300517.

References

[1] S. Brandt. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, andWhat Else?. In R. López de Mantáras et al. eds. In Proc. ECAI-2004, pp. 298-302. IOS Press, 2004.

[2] F. Baader, D. Calvanese, D.L. McGuin-

ness, D. Nardi, P.F. Patel-Schneider, eds. Description Logic Handbook, Cambridge University Press, 2002

[3] T. Calvo, A. Kolesárová, M. Komorníková, R. Mesiar. Aggregation operators: properties, classes and construction methods. In Aggregation operators: new trends and applications, R. Mesiar et al. eds. Physica-Verlag Heidelberg , 2002, pp. 3-104

[4] R. Fagin, Combining fuzzy information from multiple systems, J. Comput. System Sci. 58, 1999, 83-99

[5] D. Fensel, U. Keller, H. Lausen, A. Polleres, I. Toma. WWW or What is Wrong with Web service discovery, Position Paper for the Workshop on Web Services, Innsbruck, Austria, June 2005

[6] J. Galindo, A. Urrutia, M. Piatini. Fuzzy Databases. Idea GP, Hershey 2006

[7] FLSW - Fuzzy logic and the semantic web, E. Sanchez ed. Capturing Intelligence Series, 1, Elsevier 2006

[8] P. Hitzler, R. Studer, and Y. Sure, Description Logic Programs: A Practical Choice For the Modeling of Ontologies. In FOMI'05, Verona, Italy, June 2005.

[9] I. F. Ilyas, R. Shah, W. G. Aref, J. S. Vitter, A. K. Elmagarmid. Rank-aware query optimization. In SIGMOD 2004, ACM 2004, 203 - 214

[10] E. Jencušová, J. Jirásek: Formal Methods of Analysis of Security Protocols, Tatra Mt. Math. Publ. 25 (2002), p. 1-10

[11] M. Kifer, V. S. Subrahmanian, "Theory of generalized annotated logic programming and its applications", J. Logic Programming, 12 (1992) pp 335–367

[12] S. Krajči, R. Lencses, P. Vojtáš, A comparison of fuzzy and annotated logic programming, Fuzzy Sets and Systems, 144, 173-192 (2004)

[13] R. Kuesters, R. Molitor. Approximating most specific concepts in description logic with existential restrictions. In KI 2001, F. Baader, G. Brewka, T. Eiter eds. LNAI 2174, 33-47

[14] P. Návrat, M. Bieliková, V. Rozinajová. Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In CompSysTech 2005, Varna 2005

[15] D. Papadias, Y. Tao, G. Fu, B. Seeger. Progressive skyline computation in database systems. ACM Transactions on Database Systems, 30(2005) 41 - 82

[16] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, I. Horrocks. The Fuzzy Description Logic f-SHIN. In [19], 67–76

[17] Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence and Research 14 (2001) 137166

[18] Straccia, U.: Towards a Fuzzy Description Logic for the Semantic Web, In ESWC-05, A. Gómez-Pérez et al. eds. LNCS 3532 Springer 2005, 167–181

[19] URSW - Uncertainty Reasoning for the Semantic Web, an ISWC 2005 workshop, P. C. G. Costa et al. eds.

[20] P. Vojtáš, T. Horváth, S. Krajči, R. Lencses, An ILP model for a monotone graded classification problem, Kybernetika 40(3), 317-332 (2004).

[21] P. Vojtáš, Fuzzy logic programming, Fuzzy Sets and Systems, 124(3), 361-370 (2001)