Comparison of Scoring and Order Approach in Description Logic $\mathcal{EL}(\mathcal{D})$

Veronika Vaneková[1] and Peter Vojtáš[2]

[1] Institute of Computer Science, Pavol Jozef Šafárik University
Košice, Slovakia
veronika.vanekova@upjs.sk

[2] Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
peter.vojtas@mff.cuni.cz

Abstract. In this paper we study scoring and order approach to concept interpretation in description logics. Only concepts are scored/ordered, roles remain crisp. The concepts in scoring description logic are fuzzified, while the concepts in order description logic are interpreted as preorders on the domain. These description logics are used for preferential user-dependent search of the best instances. In addition to the standard constructors we add top-k retrieval and aggregation of user preferences. We analyze the relationship between scoring and order concepts and we introduce a notion of order-preserving concept constructors.

1 Introduction

The main motivation of our research is a large amount of data available on the web. It is the effort of Semantic Web community to use ontologies and description logics (DLs) for expressing knowledge about this data. One of the approaches to tackle with this problem are fuzzy description logics (fuzzy ontologies). The connection of description logic and fuzzy set theory appears to be suitable for many areas of Soft Computing and Semantic Web.

One interesting application of fuzzy DLs is representation of user preferences. Users naturally express their preferences in a vague, imprecise way (e.g. "I want to buy a cheap and fast car"). It is possible to handle such vague requirements with fuzzy sets. Moreover, users often need only top-k best answers, ordered by their specific preferences. Therefore we use modified top-k algorithm [6] for user-dependent search of k best objects.

We explore a specific problem when the knowledge base has a simple structure but contains a large number of individuals. We choose the description logic traditionally called $\mathcal{EL}$ ([1,2]), which allows only concept conjunction and existential restrictions to define complex concepts. Then we add fuzzy concepts to be able to represent user preference. Thus an individual which belongs to a preferential concept `good_car` to degree 0.9 is preferred more than an individual with a membership degree 0.5. Fuzzy membership values are handled by a concrete domain $\mathcal{D}$ (inspired by [11], see also [8] chapter 6 for an introduction to concrete

J. van Leeuwen et al. (Eds.): SOFSEM 2010, LNCS 5901, pp. 709–720, 2010.

domains). This kind of fuzziness does not affect roles, so they remain crisp. We call the resulting DL a *scoring description logic*, $s\text{-}\mathcal{EL}(\mathcal{D})$.

The specific fuzzy membership value or score does not play any role in many applications - only the order implied by the fuzzy value matters. The actual score is hidden to user in some applications, e.g. Google search. Therefore we present another DL called *order DL* $o\text{-}\mathcal{EL}(\mathcal{D})$, where concepts are interpreted as preorders of instances, and roles remain crisp. Having such a DL enables us to combine a part of a knowledge base describing user preferences as a module with the rest of the knowledge base (i.e. using an ontology).

The two description logics defined in this paper have many similarities. We can define a corresponding order concept for every scoring concept. An important question is whether complex concepts also have this feature. The main contributions of this paper are results on connections of $o\text{-}\mathcal{EL}(\mathcal{D})$ to $s\text{-}\mathcal{EL}(\mathcal{D})$.

2 Description Logic $s\text{-}\mathcal{EL}(\mathcal{D})$ with Scoring Concepts and Aggregation

This section briefly recalls $s\text{-}\mathcal{EL}(\mathcal{D})$, first published in [4]. As any other DL, $s\text{-}\mathcal{EL}(\mathcal{D})$ describes its universe of discourse (domain) using *concepts* and *roles*. Concepts can be viewed as classes of individuals (objects), while roles can be viewed as binary predicates describing various relationships among individuals. See [8] for basic DL theory.

The description logic $s\text{-}\mathcal{EL}(\mathcal{D})$ is fuzzified, however, it differs from other fuzzy DLs. It is designed especially to represent user preferences and allow preferential top-k queries. The main difference is that we use *crisp roles* to describe relationships of objects from the domain and *fuzzy (scoring) concepts* to represent vague user preferences. Allowed constructors are top concept $\top$, conjunction $C \sqcap D$ and existential restriction $\exists R.C$. A DL knowledge base consists of a TBox with complex concept definitions $C \sqsubseteq D$ and $C \equiv D$ and an ABox with assertions about individuals (role assertions $(a, c) : R$ and concept assertions $\langle a : C, t\rangle$, where t is a truth value). We use a finite set of truth values $TV_n = \{\frac{i}{n} : i \in \{0, \dots, n\}\} \subset [0, 1]$.

Every interpretation $\mathcal{I}$ consist of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\bullet^{\mathcal{I}}$. Concepts are interpreted as fuzzy sets of elements from the domain (see Table 1, row 1), roles are interpreted as crisp relations $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Interpretations of complex concepts can be seen on Table 1, where A, C, D are concept names, R is a role, u is a concrete role and a, b are individuals.

An interpretation $\mathcal{I}$ is a model of TBox definition $C \sqsubseteq D$ iff $\forall x \in \Delta^{\mathcal{I}} : C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$, definition $C \equiv D$, iff $\forall x \in \Delta^{\mathcal{I}} :\ C^{\mathcal{I}}(x) = D^{\mathcal{I}}(x)$, ABox concept assertion $\langle a : C, t\rangle$ iff $C^{\mathcal{I}}(a) \geq t$ and role assertions $(a, c) : R$ iff $(a, c) \in R^{\mathcal{I}}$.

The interpretation of existential quantifier $\exists R.C$ is similar to other fuzzy DLs (like $\sup_{b \in \Delta^{\mathcal{I}}} \min\{R^{\mathcal{I}}(a, b), C^{\mathcal{I}}(b)\}$ in [11]), except that we use crisp roles, so the value of $R^{\mathcal{I}}(a, b)$ is always 0 or 1. Apart from standard $\mathcal{EL}$ constructors allowed in the TBox, we add concrete domain predicates P, aggregation $@_U$ and top-k constructor. They are described more closely throughout the rest of this section.

Table 1. Syntax and Semantics of $s\text{-}\mathcal{EL}(\mathcal{D})$

Syntax	Semantics
A	$A^{\mathcal{I}} : \Delta^{\mathcal{I}} \longrightarrow TV_n$
$\top$	$\top^{\mathcal{I}} : \Delta^{\mathcal{I}} \longrightarrow \{1\}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = \min\{C^{\mathcal{I}}(a), D^{\mathcal{I}}(a)\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(b) : (a,b) \in R^{\mathcal{I}}\}$
$\exists u.P$	$(\exists u.P)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{D}}} \{P(b) : (a,b) \in u^{\mathcal{I}}\}$
top-k(C)	$\text{top-k}(C)^{\mathcal{I}}(a) = \begin{cases} C^{\mathcal{I}}(a), & \text{if } \lvert b \in \Delta^{\mathcal{I}} : C^{\mathcal{I}}(a) < C^{\mathcal{I}}(b) \rvert < k \\ 0, & \text{otherwise} \end{cases}$
$@_U(C_1, \ldots, C_m)$	$@_U(C_1, \ldots, C_m)^{\mathcal{I}}(a) = \frac{\sum_{i=1}^{m} w_i C^{\mathcal{I}}(a)}{\sum_{i=1}^{m} w_i}$

Example 1. Let the knowledge base contain data from the used car sales. We have a concept `car` and roles `has_horsepower`, `has_price`, `has_mileage`. Facts in the ABox are transfered from a (crisp) relational database, so the membership values of both individuals `Audi_A3` and `Mercedes_Benz_280` in the fuzzy concept `car` is equal to 1.

$\langle \texttt{Audi\_A3} : \texttt{car}, 1\rangle$
$(\texttt{Audi\_A3}, 7900) : \texttt{has\_price}$
$(\texttt{Audi\_A3}, 73572) : \texttt{has\_mileage}$
$(\texttt{Audi\_A3}, 110) : \texttt{has\_horsepower}$
$\langle \texttt{Mercedes\_Benz\_280} : \texttt{car}, 1\rangle$
$(\texttt{Mercedes\_Benz\_280}, 9100) : \texttt{has\_price}$
$(\texttt{Mercedes\_Benz\_280}, 127576) : \texttt{has\_mileage}$
$(\texttt{Mercedes\_Benz\_280}, 147) : \texttt{has\_horsepower}$

Different users may have different preferences to price, mileage and horsepower. Instead of exact preferences, we support vague concepts like `good_price`, `good_mileage`, `good_horsepower`. Interpretations of these concepts vary from user to user – for example, 20000 EUR is a good price for one user and unacceptable for another. Therefore all preference concepts will be user-specific (e.g. $\texttt{good\_price}_{U_1}$ for user U_1) and we represent them with concrete domain predicates [8].

We chose the concrete domain $\mathcal{D}$ originally defined in [11], because it contains basic trapezoidal predicates, sufficient to represent user preferences. It is defined as $\mathcal{D} = (\Delta^{\mathcal{D}}, Pred(\mathcal{D}))$, where the domain $\Delta^{\mathcal{D}} = \mathbb{R}$ and the set of predicates $Pred(\mathcal{D}) = \{lt_{a,b}(x), rt_{a,b}(x), trz_{a,b,c,d}(x), inv_{a,b,c,d}(x)\}$ contains monotone and trapezoidal fuzzy sets (unary fuzzy predicates). The interpretation of fuzzy predicates is fixed, handled by the concrete domain. Figure 1 shows $lt_{a,b}(x)$ (left trapezoidal membership function), $rt_{a,b}(x)$ (right trapezoidal), $trz_{a,b,c,d}(x)$ (trapezoidal) and $inv_{a,b,c,d}(x)$ (inverse trapezoidal) with one variable x and parameters a, b, c, d.

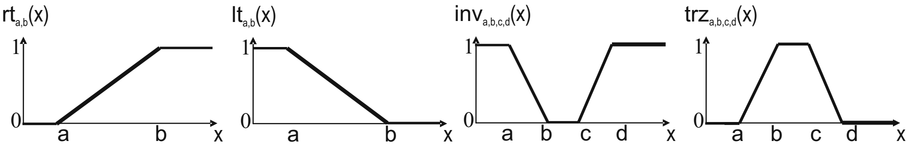

Fig. 1. Fuzzy membership functions of basic trapezoidal types

Fuzzy concrete domain $\mathcal{D}$ adds a concept constructor $\exists u.P$, where $P \in Pred^{\mathcal{D}}$ and u is a concrete role. This constructor generates a total preorder, because P has a fixed interpretation.

Example 2. We further extend the knowledge base stated above with preference concepts of user U_1.

$\texttt{good\_price}_{U_1} \equiv \exists(\texttt{has\_price}).lt_{7000,9000}$
$\texttt{good\_horsepower}_{U_1} \equiv \exists(\texttt{has\_horsepower}).rt_{100,150}$
$\texttt{good\_car}_{U_1} \equiv \texttt{car} \sqcap \texttt{good\_price} \sqcap \texttt{good\_horsepower}$

Then every model $\mathcal{I}$ of the knowledge base will satisfy the following:

$\langle \texttt{Audi\_A3} : \texttt{good\_price}_{U_1}, 0.55\rangle$
$\langle \texttt{Audi\_A3} : \texttt{good\_horsepower}_{U_1}, 0.2\rangle$
$\langle \texttt{Audi\_A3} : \texttt{good\_car}_{U_1}, 0.2\rangle$
$\langle \texttt{Mercedes\_Benz\_280} : \texttt{good\_price}_{U_1}, 0\rangle$
$\langle \texttt{Mercedes\_Benz\_280} : \texttt{good\_horsepower}_{U_1}, 0.94\rangle$
$\langle \texttt{Mercedes\_Benz\_280} : \texttt{good\_car}_{U_1}, 0\rangle$

Fuzzy conjunction is not always suitable to represent user preferences because it penalizes some objects that may be still interesting for the user. This is the case of the individual `Mercedes_Benz_280`, which has very good value of horsepower for user U_1, but the price is beyond preferred range. Instead of the conjunction, we can use aggregation to obtain overall preference value. Aggregation functions are m-ary fuzzy functions $@_U : TV_n^m \longrightarrow TV_n$, monotone in all arguments and such that $@_U(1, \ldots, 1) = 1$ and $@_U(0, \ldots, 0) = 0$. We do not require other properties such as being homogeneous, additive or Lipschitz continuous, though such functions can be more useful for the computation. Note that aggregations are a generalization of both fuzzy conjuctions and disjunctions. As such, unrestricted aggregations add too much expressive power to the language [10]. According to the paper [5], the instance problem for DLs with aggregation is polynomially decidable, provided that the subsumption of two aggregations can be decided in polynomial time.

If we use aggregation, the concept $\texttt{good\_car}_{U_1}$ from our example will be $@_{U_1}(\texttt{good\_price}_{U_1}, \texttt{good\_horsepower}_{U_1})$, where the aggregation is a weighted average $@_{U_1}(x, y) = \frac{x+2y}{3}$. Then the minimal model will be different:

$\langle \texttt{Audi\_A3} : \texttt{good\_car}, 0.32\rangle$
$\langle \texttt{Mercedes\_Benz\_280} : \texttt{good\_car}, 0.63\rangle$

Instead of standard reasoning tasks, we add a new task, *top-k retrieval*, to our description logic: for a given concept C, find k individuals with the greatest membership degrees. The value of top-k$(C)^{\mathcal{I}}(a)$ is equal to $C^{\mathcal{I}}(a)$ if a belongs to k best individuals from preference concept C (or it is preferred equally as k^{th} individual). Otherwise the value is set to 0. Note that according to the definition of top-k$(C)^{\mathcal{I}}(a)$ in Table 1, we return those elements for which the number of strictly greater elements is less than k, which includes the ties on the k^{th} position. Thus the result can possibly include more than k individuals. The original top-k algorithm [7] is non-deterministic – instead of returning all ties on the k^{th} position, it chooses some of them randomly and returns exactly k objects. We use this non-standard definition in order to make the top-k constructor deterministic.

Top-k algorithm [6] uses a preprocessing stage to generate lists of individuals ordered by role values. In the example above it would generate three ordered lists of individuals (for `has_price`, `has_mileage` and `has_horsepower`). The lists are traversed in specific order dependent on the particular fuzzy sets. The algorithm evaluates aggregation function on the fly and determines threshold value to find out if some other objects can have greater overall value than objects already processed. This algorithm has proved to be very efficient (see [6]).

3 Description Logic *o*-$\mathcal{EL}(\mathcal{D})$ with Concept Instance Ordering

Every preference concept defined in the previous section generates ordering of individuals according to the preference value. Conjunctions or aggregations are necessary to obtain overall order. This principle is similar to the decathlon rules: athletes compete in ten disciplines, each discipline is awarded with points according to scoring tables. All points are summed up to determine the final order. This is the case when all precise scores are important to determine the final score.

There are other cases when the score itself is not important. Recall another example from sport: in Formula 1, the first eight drivers gain points according to the point table (10, 8, 6, 5, 4, 3, 2, 1) regardless of their exact time, speed or headstart. The final order is also determined by summing up all points. A similar system is used in Tour de France, where riders can earn points at the end of each stage. The stages are divided into several types and each type has its own point table.

To return back to computer science, user preference is often represented as an order, partial or total. The partial order can be induced from user inputs like "object a is better than object b" or from sample set of objects rated by the user. Inductive learning of user preference from such rated set of objects means finding a linear extension of the partial order and thus be able to compare any pair of objects.

In this section we propose description logic *o*-$\mathcal{EL}(\mathcal{D})$ with concepts $C_{\leq}$ interpreted as non-strict preorders on the domain, $C_{\leq}^{\mathcal{J}}$. From a logical point of view, we interpret both concepts and roles as binary predicates. If $(a, b) \in C_{\leq}^{\mathcal{J}}$,

Table 2. Syntax and Semantics of o-$\mathcal{EL}(\mathcal{D})$

Syntax	Semantics
$A_\leq$	$A_\leq^\mathcal{J} \subseteq \Delta^\mathcal{J} \times \Delta^\mathcal{J}$
$\top_\leq$	$\Delta^\mathcal{J} \times \Delta^\mathcal{J}$
$C_\leq \sqcap D_\leq$	$\{(a_1, a_2) : (a_1, a_2) \in C_\leq^\mathcal{J} \wedge (a_1, a_2) \in D_\leq^\mathcal{J}\}$
$\exists R.C_\leq$	$\{(a_1, a_2) : \forall c_1\ (a_1, c_1) \in R^\mathcal{J}\ \exists c_2\ (a_2, c_2) \in R^\mathcal{J} : (c_1, c_2) \in C_\leq^\mathcal{J}\}$
$\exists u.P$	$\{(a_1, a_2) : \forall c_1\ (a_1, c_1) \in u^\mathcal{J}\ \exists c_2\ (a_2, c_2) \in u^\mathcal{J} : P(c_1) \leq P(c_2)\}$
$@_U(C_{\leq 1}, \ldots, C_{\leq m})$	defined below
top-k(C)	defined below

then a belongs to the concept $C_\leq$ less than b (or equally). If $C_\leq$ is a concept representing user preference, we say that b is preferred to a. Complex concepts in o-$\mathcal{EL}(\mathcal{D})$ are constructed according to Table 2 (compare with Table 1). Lowercase letters denote individuals, except for u which denotes a concrete role. Two additional non-standard constructors (aggregation and top-k constructor) are defined in the subsequent text.

A preorder is a reflexive and transitive relation. We do not need the antisymmetry condition (as is required for partial orders) because there can be two individuals that are not identical despite being equally preferred. We call such individuals *indiscernible* according to preference concept C. A preorder is *total*, if $\forall a, b \in \Delta^\mathcal{J} :\ (a, b) \in C_\leq^\mathcal{J} \vee (b, a) \in C_\leq^\mathcal{J}$ (one or both inequalities can hold).

We use the same concrete domain as in case of s-$\mathcal{EL}(\mathcal{D})$. Typical TBox definitions are $C_\leq \sqsubseteq D_\leq$ and $C_\leq \equiv D_\leq$. The ABox contains concept assertions $(a_1, a_2) : C_\leq$ and role assertions $(a, c) : R$. To distinguish the interpretations in o-$\mathcal{EL}(\mathcal{D})$ from s-$\mathcal{EL}(\mathcal{D})$, we denote the order-oriented interpretations as $\mathcal{J}$.

An interpretation $\mathcal{J}$ is a model of $C \sqsubseteq D$ iff $(a, b) \in C_\leq^\mathcal{J}$ implies $(a, b) \in D_\leq^\mathcal{J}$, and $C \equiv D$, iff $C^\mathcal{J} = D^\mathcal{J}$. $\mathcal{J}$ is a model of an ABox assertion $(a_1, a_2) : C_\leq$ iff $(a_1, a_2) : C_\leq^\mathcal{J}$. Role assertions are handled in the same way as in s-$\mathcal{EL}(\mathcal{D})$.

Top concept $\top_\leq$ is interpreted as a complete relation $\Delta^\mathcal{J} \times \Delta^\mathcal{J}$, where all individuals are equally preferred. Concept conjunction $C_\leq \sqcap D_\leq$ often produces partial preorder, even if $C_\leq$ and $D_\leq$ are total preorders. According to the *order-extension principle*, it is possible to extend $(C_\leq \sqcap D_\leq)^\mathcal{J}$ to a total preorder. However, this extension does not have to be unique. Sometimes it is more convenient to use aggregation $@_U$ instead of concept conjunction, especially when we consider a conjunction of more than two concepts.

The semantics of $\exists R.C_\leq$ is chosen to be analogous with s-$\mathcal{EL}(\mathcal{D})$. Imagine that the individual a_1 is connected with c_1, c_2, c_3 (values of role R), while a_2 is connected with c_4, c_5, c_6. In the scoring case, we would simply take the supremum of $C^\mathcal{I}(c_1), C^\mathcal{I}(c_2), C^\mathcal{I}(c_3)$ as the fuzzy value of $\exists R.C^\mathcal{I}(a_1)$ and analoguously for $\exists R.C^\mathcal{I}(a_2)$. Then we just compare the suprema. To simulate the "supremum" in the ordering case, we must define that for every c_i connected with a_1 there exists a better c_j (with respect to the preference concept $C_\leq$) connected with a_2 via role R.

For every $@_U$ with arity m and for every m-tuple of order concepts $C_{\leq_j} \subseteq \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$ aggregation $@_U^{\mathcal{J}}(C_{\leq_1}, \ldots, C_{\leq_m}) \subseteq \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$ is a partial preorder. If $(a,b) \in C_{\leq_j}^{\mathcal{J}}$ for every $j = 1, \ldots, m$, then $(a,b) \in @_U^{\mathcal{J}}(C_{\leq_1}, \ldots, C_{\leq_m})$. We define aggregation similar to Formula 1 rules. First of all, it is necessary to define the *level* of instance a in the interpretation of concept C. It is the biggest possible length of a sequence such that the first element is a and every following element is strictly greater than its predecessor.

$$level(C, a, \mathcal{J}) = \max_{l \in \mathbf{N}} \{l : \exists b_1, \ldots, b_l \in \Delta^{\mathcal{J}}\ \forall i \in \{1, \ldots, l-1\}(b_i, b_{i+1}) \in C^{\mathcal{J}} \wedge (b_{i+1}, b_i) \notin C^{\mathcal{J}} \wedge b_1 = a\}$$

Next we define a *scoring table* for the aggregation, which is a finite strictly decreasing sequence $score_{@_U}(score_1, \ldots, score_m)$ like (10, 8, 6, 5, 4, 3, 2, 1). The differences bet ween adjacent elements are also decreasing, but not strictly. The pair (a,b) belongs to aggregation $@_U^{\mathcal{J}}(C_{\leq_1}, \ldots, C_{\leq_m})$ if

$$\sum_{j=1}^{m} score_{level(C_{\leq_j}, a, \mathcal{J})} \leq \sum_{j=1}^{m} score_{level(C_{\leq_j}, b, \mathcal{J})}.$$

This means that we find the level of individual a in every preference concept $C_{\leq_j}^{\mathcal{J}}$, then we determine the corresponding scores for these levels and sum up all the scores. If the individual b has better levels in the preference concepts $C_{\leq_j}^{\mathcal{J}}$ than individual a, it will also have a higher sum of all scores.

It is also straightforward to define top-k queries. Let $C_a = \{c \in \Delta^{\mathcal{J}} : (a,c) \in C_{\leq}^{\mathcal{J}} \wedge (c,a) \notin C_{\leq}^{\mathcal{J}}\}$ be a set of individuals strictly greater than a in ordering concept $C_{\leq}$. Then $(a,b) \in$ top-k$(C_{\leq})^{\mathcal{J}}$, iff:

1. $(a,b) \in C_{\leq}^{\mathcal{J}}$ or
2. $|C_a| \geq k$

If $C_{\leq}$ was a total preorder, then top-k$(C_{\leq})$ will be also total. Top-k constructor preserves the original order of the first k individuals, including the ties. Note that the first k individuals often occupy less than k levels because some of them are ties. Concerning the ties on the last included level (not necessarilly the k-th level), we can either choose only some of them to fill up the needed amount of elements, or we can return them all. We choose the latter possibility, even if we end up with more than k elements in the result, because it makes our definition deterministic. If some element a has more than k strictly greater elements in $C_{\leq}^{\mathcal{J}}$, so that it is beyond the last included level (see condition 2), it is made lower or equal to all other elements, which moves it to the last level in top-k$(C_{\leq})^{\mathcal{J}}$.

Example 3. We transform the knowledge base from the previous section:

(Audi_A3, Mercedes_Benz_280) : car
(Mercedes_Benz_280, Audi_A3) : car
(Audi_A3, 7900) : has_price
(Audi_A3, 110) : has_horsepower
(Mercedes_Benz_280, 9100) : has_price

$(\texttt{Mercedes\_Benz\_280}, 147) : \texttt{has\_horsepower}$
$\texttt{good\_price}_{U_1} \equiv \exists(\texttt{has\_price}).lt_{7000,9000}$
$\texttt{good\_horsepower}_{U_1} \equiv \exists(\texttt{has\_horsepower}).rt_{100,150}$
$\texttt{good\_car}_{U_1} \equiv \texttt{car} \sqcap \texttt{good\_price}_{U_1} \sqcap \texttt{good\_horsepower}_{U_1}$

Every model $\mathcal{J}$ of the knowledge base will satisfy the following:

$(\texttt{Mercedes\_Benz\_280}, \texttt{Audi\_A3}) : \texttt{good\_price}_{U_1}$
$(\texttt{Audi\_A3}, \texttt{Mercedes\_Benz\_280}) : \texttt{good\_horsepower}_{U_1}$

The latter assertion is satisfied because $\forall c_1$ $(\texttt{Audi\_A3}, c_1) \in \texttt{has\_horsepower}^{\mathcal{J}}$ $\exists c_2 \in \Delta^{\mathcal{J}}$ $(\texttt{Mercedes\_Benz\_280}, c_2) \in \texttt{has\_horsepower}^{\mathcal{J}} : (c_1, c_2) \in rt^{\mathcal{J}}_{100,150}$. We have only one possibility $c_1 = 110$ and $c_2 = 147$ and moreover $(110, 147) \in rt^{\mathcal{J}}_{100,150}$.

Note that neither the tuple $(\texttt{Audi\_A3}, \texttt{Mercedes\_Benz\_280})$, nor the tuple $(\texttt{Mercedes\_Benz\_280}, \texttt{Audi\_A3})$ belongs to $\texttt{good\_car}^{\mathcal{J}}_{U_1}$ in every model $\mathcal{J}$. This is caused by the ambiguity in concept conjunctions (because the interpretation of the concept $\texttt{good\_car}_{U_1}$ is a partial preorder). This shows a necessity to use aggregations instead of concept conjunctions. Let us define the scoring table for aggregation $@_{U_1}$ as $(3, 2, 1)$ and the preferential concept $\texttt{good\_car}_{U_1}$ as $@_{U_1}(\texttt{good\_price}_{U_1}, \texttt{good\_horsepower}_{U_1})$. Individual Audi_A3 has the first place in concept $\texttt{good\_horsepower}_{U_1}$, while Mercedes_Benz_280 is first in the concept $\texttt{good\_price}_{U_1}$. The result is that both individuals gain five points in total (three for the first place and two for the second place) and every interpretation must satisfy both:

$(\texttt{Mercedes\_Benz\_280}, \texttt{Audi\_A3}) : \texttt{good\_car}_{U_1}$
$(\texttt{Audi\_A3}, \texttt{Mercedes\_Benz\_280}) : \texttt{good\_car}_{U_1}$

4 Relationship between Scoring and Order Approach

Definitions for $s\text{-}\mathcal{EL}(\mathcal{D})$ and $o\text{-}\mathcal{EL}(\mathcal{D})$ are much similar, but the two logics are not equivalent. At the first sight, it is obvious that $o\text{-}\mathcal{EL}(\mathcal{D})$ drops exact membership degrees, thus it loses the ability to express some features of $s\text{-}\mathcal{EL}(\mathcal{D})$. If we have a "constant" fuzzy concept $C^{\mathcal{I}}(a) = w \in TV_n$ for every $a \in \Delta^{\mathcal{I}}$, the corresponding order concept in $o\text{-}\mathcal{EL}(\mathcal{D})$ will be $C^{\mathcal{J}}_{\leq} = \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$, regardless of the value w. Similarly, if $D^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(b)$, the corresponding order concept contains the pair $(a, b) \in D^{\mathcal{J}}_{\leq}$, but we lose information about the difference $D^{\mathcal{I}}(b) - D^{\mathcal{I}}(a)$.

If we compare a scoring concept C with ordering concept $C_{\leq}$, we are concerned about the order of individuals. It is straightforward to define corresponding order-preserving concept $A_{\leq}$ for every primitive concept A and for any interpretation $A^{\mathcal{I}}$. We define $(a, b) \in A^{\mathcal{J}}_{\leq}$ iff $A^{\mathcal{I}}(a) \leq A^{\mathcal{I}}(b)$. Concept constructors should also preserve order of individuals. We start from a scoring concept A, transform it to a corresponding ordering concept $A_{\leq}$, use constructors (let us denote a generic constructor as $m(A)$, $m(A_{\leq})$) on both concepts and finally compare order of individuals in the results. If $m(A_{\leq})$ is a partial preorder, there are many possible total extensions.

Let $(a,b) \in A_{\leq}^{\mathcal{J}}$ iff $A^{\mathcal{I}}(a) \leq A^{\mathcal{I}}(b)$. Constructor $m(A)$ is *order-preserving* if there exists a linear extension $m(A_{\leq})'$ of $m(A_{\leq})$ such that $((a,b) \in m(A_{\leq})'^{\mathcal{J}}$ iff $m(A)^{\mathcal{I}}(a) \leq m(A)^{\mathcal{I}}(b))$. Note that the concrete domain $\mathcal{D}$ is already defined in such a way that $\exists u.P$ is order-preserving.

Lemma 1. *Existential quantification is order-preserving.*

Proof. Let $C_{\leq}$ be order-preserving concept for C. Let $(a_1, a_2) \in \exists R.C_{\leq}^{\mathcal{J}}$. According to the definition, $\forall c_1\ (a_1, c_1) \in R^{\mathcal{J}}\ \exists c_2\ (a_2, c_2) \in R^{\mathcal{J}} : (c_1, c_2) \in C_{\leq}^{\mathcal{J}}$. We know that the interpretation of roles is the same and C is order-preserving, thus $\forall c_1\ (a_1, c_1) \in R^{\mathcal{I}}\ \exists c_2\ (a_2, c_2) \in R^{\mathcal{I}} : C^{\mathcal{I}}(c_1) \leq C^{\mathcal{I}}(c_2)$. The same inequality holds for suprema: $\sup_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_1) : (a_1, c_1) \in R^{\mathcal{I}}\} \leq \sup_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_2)|\ (a_2, c_2) \in R^{\mathcal{I}}\}$ Therefore $(\exists R.C)^{\mathcal{I}}(a_1) \leq (\exists R.C)^{\mathcal{I}}(a_2)$.

For the reversed implication, suppose that $(\exists R.C)^{\mathcal{I}}(a_1) \leq (\exists R.C)^{\mathcal{I}}(a_2)$, and from the definition also $\sup_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_1) : (a_1, c_1) \in R^{\mathcal{I}}\} \leq \sup_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_2) : (a_2, c_2) \in R^{\mathcal{I}}\}$. Since the set of truth values is finite, the supremum must belong to the set. Towards the contradiction, suppose that $\exists c_1\ (a_1, c_1) \in R^{\mathcal{I}}\ \forall c_2\ (a_2, c_2) \in R^{\mathcal{I}} : C^{\mathcal{I}}(c_1) > C^{\mathcal{I}}(c_2)$. Then $C^{\mathcal{I}}(c_1)$ is upper bound of the set and it is greater than the maximum $C^{\mathcal{I}}(c_1) > \max_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_2) : (a_2, c_2) \in R^{\mathcal{I}}\} \geq \max_{b \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(c_1) : (a_1, c_1) \in R^{\mathcal{I}}\} \geq C^{\mathcal{I}}(c_1)$, which is a contradiction. Thus $\forall c_1\ (a_1, c_1) \in R^{\mathcal{I}}\ \exists c_2\ (a_2, c_2) \in R^{\mathcal{I}} :\ C^{\mathcal{I}}(c_1) \leq C^{\mathcal{I}}(c_2)$. As C is order-preserving concept, we gain $(a_1, a_2) \in \exists R.C_{\leq}^{\mathcal{J}}$. □

Note that in case of fuzzy s-$\mathcal{EL}(\mathcal{D})$, we define $\sup \emptyset = 0$. In case of o-$\mathcal{EL}(\mathcal{D})$, if no individual is connected to a_1 with role R, then $(a_1, a_2) \in \exists R.C_{\leq}^{\mathcal{J}}$, so it yields correct inequalities for $0 \leq x$ and $0 \leq 0$.

Lemma 2. *Constructor top-k is order-preserving.*

Proof. Let us suppose that top-k$(C)^{\mathcal{I}}(a_1) \leq$ top-k$(C)^{\mathcal{I}}(a_2)$. Note that the condition $\left|c \in \Delta^{\mathcal{I}} : C^{\mathcal{I}}(a) < C^{\mathcal{I}}(c)\right| < k$ is equivalent to $|C_a| < k$ because C is order-preserving. Since top-k$(C)^{\mathcal{I}}(x)$ can be either 0 or $C^{\mathcal{I}}(x)$, we have three possibilities:

- *case 1)* top-k$(C)^{\mathcal{I}}(a_1) =$ top-k$(C)^{\mathcal{I}}(a_2) = 0$
- *case 2)* $0 =$ top-k$(C)^{\mathcal{I}}(a_1) \leq$ top-k$(C)^{\mathcal{I}}(a_2) = C^{\mathcal{I}}(a_2)$
- *case 3)* $C^{\mathcal{I}}(a_1) =$ top-k$(C)^{\mathcal{I}}(a_1) \leq$ top-k$(C)^{\mathcal{I}}(a_2) = C^{\mathcal{I}}(a_2)$

Case 1 and 2: From the definition and the equivalence of conditions above we have $|C_{a_1}| \geq k$. This is the condition 2 from the definition of top-k$(C_{\leq})$, and thus $(a_1, a_2) \in$ top-k$(C_{\leq})^{\mathcal{J}}$.

Case 3: $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2)$ means that $(a_1, a_2) \in C^{\mathcal{J}}$, which is the condition 1 from the definition of top-k$(C_{\leq})$, and thus also $(a_1, a_2) \in$ top-k$(C_{\leq})^{\mathcal{J}}$.

Now let $(a_1, a_2) \in \text{top-k}(C_{\leq})^{\mathcal{J}}$. This can be a consequence of the condition 1 or 2.

Let condition 2 hold – we know that $|C_{a_1}| \geq k$ and from the equivalence of conditions $\left|c \in \Delta^{\mathcal{I}} : C^{\mathcal{I}}(a_1) < C^{\mathcal{I}}(c)\right| \geq k$. From the definition of top-k(C) follows that top-k $(C)^{\mathcal{I}}(a_1) = 0$, so it will always be less or equal than top-k $(C)^{\mathcal{I}}(a_2)$.

Let condition 1 hold and let $|C_{a_1}| < k$ (otherwise we could apply the proof above). Because a_2 is greater than a_1 in preorder $C_{\leq}^{\mathcal{J}}$, the set of greater elements will also have cardinality less than k. Thus top-k$(C)^{\mathcal{I}}(a_1) = C^{\mathcal{I}}(a_1)$ and top-k$(C)^{\mathcal{I}}(a_2) = C^{\mathcal{I}}(a_2)$ and moreover $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2)$, which yields top-k$(C)^{\mathcal{I}}(a_1) \leq$ top-k$(C)^{\mathcal{I}}(a_2)$. □

Constructor $C_{\leq} \sqcap D_{\leq}$ produces partial preorders. Because of the minimum function in $(C \sqcap D)^{\mathcal{I}}$, we cannot model this constructor exactly in $o\text{-}\mathcal{EL}(\mathcal{D})$. There is no way of comparing elements without fuzzy degrees in two different preorders.

Lemma 3. *Concept conjunction is order-preserving.*

Proof. Let $(C \sqcap D)^{\mathcal{I}}(a_1) \leq (C \sqcap D)^{\mathcal{I}}(a_2)$. According to the definition of $C \sqcap D$, $\min\{C^{\mathcal{I}}(a_1), D^{\mathcal{I}}(a_1)\} \leq \min\{C^{\mathcal{I}}(a_2), D^{\mathcal{I}}(a_2)\}$. Let us suppose that $C^{\mathcal{I}}(a_1) = \min\{C^{\mathcal{I}}(a_1), D^{\mathcal{I}}(a_1)\}$ (the other case is analoguous). Then $C^{\mathcal{I}}(a_1)$ must be on the first place and we have six possibilities how to order all the values:

1. $C^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_2)$
2. $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2)$
3. $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_1)$
4. $C^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2) \leq C^{\mathcal{I}}(a_2)$
5. $C^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2)$
6. $C^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2) \leq C^{\mathcal{I}}(a_2) \leq D^{\mathcal{I}}(a_1)$

In cases 1, 2 or 4 we are done, because both $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2)$ and $\leq D^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2)$ hold and we have also $(a_1, a_2) \in (C_{\leq} \sqcap D_{\leq})^{\mathcal{J}}$. In cases 3, 5, 6 neither (a_1, a_2) nor (a_2, a_1) belong to $(C_{\leq} \sqcap D_{\leq})^{\mathcal{J}}$. We define the extension X to contain the tuple (a_1, a_2). All tuples added this way agree with order induced by $(C \sqcap D)^{\mathcal{I}}$. The extension X is a total preorder, so it must be reflexive, transitive and $\forall a, b \in \Delta^{\mathcal{J}} : ((a, b) \in X \vee (b, a) \in X)$. Because $(C \sqcap D)^{\mathcal{I}}$ is also a total preorder and all inequalities from $(C \sqcap D)^{\mathcal{I}}$ hold also in X, we only have to check whether X contains any extra tuples from $(C_{\leq} \sqcap D_{\leq})^{\mathcal{J}}$ that could be in conflict with $(C \sqcap D)^{\mathcal{I}}$.

Let $(a_1, a_2) \in (C_{\leq} \sqcap D_{\leq})^{\mathcal{J}}$. From the definition of concept conjunction, $(a_1, a_2) \in C_{\leq}^{\mathcal{J}} \wedge (a_1, a_2) \in D_{\leq}^{\mathcal{J}}$. Concepts C, D are order-preserving, hence $C^{\mathcal{I}}(a_1) \leq C^{\mathcal{I}}(a_2) \wedge D^{\mathcal{I}}(a_1) \leq D^{\mathcal{I}}(a_2)$. The same inequality holds for minimum, $\min\{C^{\mathcal{I}}(a_1), D^{\mathcal{I}}(a_1)\} \leq \min\{C^{\mathcal{I}}(a_2), D^{\mathcal{I}}(a_2)\}$, which means $(C \sqcap D)^{\mathcal{I}}(a_1) \leq (C \sqcap D)^{\mathcal{I}}(a_2)$. Thus X is a linear extension of $(C_{\leq} \sqcap D_{\leq})^{\mathcal{J}}$ and preserves ordering of $(C \sqcap D)^{\mathcal{I}}$. □

Note that aggregations are defined differently for $o\text{-}\mathcal{EL}(\mathcal{D})$ and $s\text{-}\mathcal{EL}(\mathcal{D})$, so we do not address their relationship here.

5 Related Work

There is a considerable effort concerning the connection of tractable description logics with top-k algorithm. Papers [9,16,17] use DL-Lite, a tractable DL with constructors $\exists R$, $\exists R^{-}$, $C_1 \sqcap C_2$, $\neg B$ and functional property axioms, together with top-k retrieval. Also DL $\mathcal{EL}$ is a subject of intensive research, in order to enhance the language without losing the tractability (see [3,14,15]).

The notion of instance ordering within description logics appeared in [12]. This paper defines crisp DL $\mathcal{ALCQ}(\mathcal{D})$ with special *ordering descriptions* that can be used to index and search a knowledge base. The paper [13] presents $\mathcal{ALC}_{fc}$, a fuzzy DL with *comparison concept constructors*, where it is possible to define e.g. a concept of very cheap cars (with fuzzy degree of "cheap" over some specified value), or cars that are more economy than strong. However, all of the mentioned papers use the classical (crisp or fuzzy) concept interpretation. To the best of our knowledge, there is no other work concerning interpreting concepts as preorders.

We already studied $\mathcal{EL}(\mathcal{D})$ with fuzzified concepts in [4]. We suggested the shift towards ordering approach, but the paper did not specify details of o-$\mathcal{EL}(\mathcal{D})$, nor the relationship between scoring and ordering description logic. In the paper [18], we proposed a basic reasoning algorithm for o-$\mathcal{EL}(\mathcal{D})$.

6 Conclusion

User preference is often represented as an order of objects. We show that it is possible to omit fuzzy scores (membership degrees) in description logics and to interpret concepts as preorders of the domain. We adopt the order-oriented approach for the standard concept constructors like existential restriction, concept conjunction and concrete domain predicates. We add extra constructors $@_U$ for aggregation and top-k for the retrieval of k best individuals from the concept. The resulting description logic is called o-$\mathcal{EL}(\mathcal{D})$. We show that the constructors in o-$\mathcal{EL}(\mathcal{D})$ preserve the order of individuals induced by fuzzy scores in s-$\mathcal{EL}(\mathcal{D})$. DL o-$\mathcal{EL}(\mathcal{D})$ is especially suited for user preference modelling, but it has also some limitations, e.g. it is difficult to adapt some classical reasoning problems to ordering case. As a part of our future research, we want to improve the reasoning algorithm for instance problem in o-$\mathcal{EL}(\mathcal{D})$ [18] and to devise a reasoning algorithm for subsumption of two order concepts.

References

1. Brandt, S.: Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and - What else? In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, pp. 298–302. IOS Press, Amsterdam (2004)
2. Baader, F., Lutz, C., Suntisrivaraporn, B.: Is Tractable Reasoning in Extensions of the Description Logic EL Useful in Practice? In: Proceedings of the Methods for Modalities Workshop, M4M 2005 (2005)

3. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope Further. In: Proceedings of the Workshop on OWL: Experiences and Directions, OWLED 2008 (2008)
4. Vaneková, V., Vojtáš, P.: A Description Logic with Concept Instance Ordering and Top-k Restriction. In: Information Modelling and Knowledge Bases XX. Frontiers in Artificial Intelligence and Applications, vol. 190, pp. 139–153. IOS Press, Amsterdam (2009)
5. Vojtáš, P.: A Fuzzy EL Description Logic with Crisp Roles and Fuzzy Aggregation for Web Consulting. In: Information Processing and Management under Uncertainty (IPMU), pp. 1834–1841. Éditions EDK, Paris (2006)
6. Gurský, P., Vojtáš, P.: On Top-k Search with No Random Access Using Small Memory. In: Atzeni, P., Caplinskas, A., Jaakkola, H. (eds.) ADBIS 2008. LNCS, vol. 5207, pp. 97–111. Springer, Heidelberg (2008)
7. Fagin, R., Lotem, A., Naor, M.: Optimal Aggregation Algorithms for Middleware. In: PODS 2001: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (2001); Journal of Computer and System Sciences 66(4), 614–656 (2001)
8. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, D.F. (eds.): Description Logic Handbook. Cambridge University Press, Cambridge (2002)
9. Straccia, U.: Towards Top-k Query Answering in Description Logics: the Case of DL-Lite. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 439–451. Springer, Heidelberg (2006)
10. Baader, F., Sattler, U.: Description Logics with Aggregates and Concrete Domains. Information Systems 28(8), 979–1004 (2003)
11. Straccia, U.: Fuzzy $\mathcal{ALC}$ with Fuzzy Concrete Domains. In: Proceedings of the 2005 International Workshop on Description Logics (DL 2005), vol. 147, pp. 96–103. CEUR Workshop Proceedings (2005)
12. Pound, J., Stanchev, L., Toman, D., Weddell, G.E.: On Ordering and Indexing Metadata for the Semantic Web. In: Proceedings of the 21st International Workshop on Description Logics (DL-2008). CEUR Workshop Proceedings, vol. 353 (2008)
13. Kang, D., Xu, B., Lu, J., Li, Y.: Reasoning for Fuzzy Description Logic with Comparison Expressions. In: Proceedings of the 2006 International Workshop on Description Logics (DL 2006). CEUR Workshop Proceedings, vol. 189 (2006)
14. Stoilos, G., Stamou, G., Pan, J.Z.: Classifying Fuzzy Subsumption in Fuzzy-EL+. In: Proceedings of the 21st International Workshop on Description Logics (DL 2008). CEUR Workshop Proceedings, vol. 353 (2008)
15. Mailis, T., Stoilos, G., Simou, N., Stamou, G.: Tractable Reasoning Based on the Fuzzy EL++ Algorithm. In: Proceedings of the Fourth International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008). CEUR Workshop Proceedings, vol. 423 (2008)
16. Straccia, U.: Answering Vague Queries in Fuzzy DL-Lite. In: Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 2238–2245 (2006)
17. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E., Taylor, S.: Scalable Querying Service over Fuzzy Ontologies. In: Proceedings of the 17th International World Wide Web Conference (WWW 2008), pp. 575–584. ACM, New York (2008)
18. Vaneková, V., Vojtáš, P.: Order-Oriented Reasoning in Description Logics. In: Proceedings of 6th Atlantic Web Intelligence Conference (AWIC 2009). Advances in Intelligent and Soft Computing, vol. 67. Springer, Heidelberg (to appear, 2010)