Variable-Strength Conditional Preferences for Ranking Objects in Ontologies

Thomas Lukasiewicz [1, 2] and Jörg Schellhase [3]

[1] DIS, Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Rome, Italy
lukasiewicz@dis.uniroma1.it

[3] Fachgebiet Wirtschaftsinformatik, Universität Kassel
Nora-Platiel-Straße 4, D-34127 Kassel, Germany
schellhase@wirtschaft.uni-kassel.de

Abstract. We introduce conditional preference bases as a means for ranking objects in ontologies. Conditional preference bases consist of a description logic knowledge base and a finite set of variable-strength conditional preferences. They are inspired by Goldszmidt and Pearl's approach to default reasoning from conditional knowledge bases in System Z^+. We define a notion of consistency for conditional preference bases, and show how consistent conditional preference bases can be used for ranking objects in ontologies. We also provide algorithms for computing the rankings. To give evidence of the usefulness of this approach in practice, we describe an application in the area of literature search.

1 Introduction

In their seminal works [34,33], Poole and Smyth deal with the problem of matching instances against models of instances, which are both described at different levels of abstraction and at different levels of detail, using qualitative probability theory. Informally, such problems can be described as follows. Given an instance I and a model of instances M, compute the qualitative probability that the instance I is matching the model M (that is, of I given M). For example, in a geological exploration domain, we may want to determine whether there might be gold in an area. In this case, an instance I may be given by the description of an area, while a model M may be given by a description of areas where gold can be found, and the qualitative probability that I is matching M describes the likelihood that gold may be found in I.

In this paper, we continue this line of research. A serious drawback of the above works [34,33] on matching instances against models of instances is that they only allow for expressing simple preferences of the form "property α is preferred over property $\neg\alpha$ with strength s" in models of instances. In particular, they do not allow for conditional preferences such as "generally, in the context ϕ, property α is preferred over property $\neg\alpha$ with strength s". In this paper, we try to fill this gap. We present a

[2] Alternate address: Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria; e-mail: lukasiewicz@kr.tuwien.ac.at.

formalism for ranking objects in description logics that allows for expressing such conditional preferences in models of instances. In a companion paper [30], we present a generalization of this formalism for matchmaking in description logics.

Like Poole and Smyth's work [34,33], the ranking formalism in this paper is also based on qualitative probabilities. Differently from Poole and Smyth's work [34,33], however, it requires a technically more involved way of computing qualitative probabilities, since our language for encoding models of instances is more expressive. We especially have to suitably handle *variable-strength conditional preferences*, which are the above statements "generally, in the context ϕ, property α is preferred over property $\neg\alpha$ with strength s" (also called *variable-strength conditional desires* [36]). They bear close similarity to *variable-strength defaults* of form "generally, if ϕ then α with strength s" in default reasoning from conditional knowledge bases (see Section 7).

In this paper, we define a formal semantics for variable-strength conditional preferences, which is based on a generalization of Goldszmidt and Pearl's default entailment in System Z^+ [22]. We focus on the problem of ranking objects against a description of objects. Since we are especially interested in the Semantic Web as the main application context, we assume that objects and descriptions of objects are expressed in the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL, respectively [23].

The Semantic Web [6,17] aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind it are to add a machine-readable meaning to Web pages, to use ontologies for a precise definition of shared terms in Web resources, to make use of KR technology for automated reasoning from Web resources, and to apply cooperative agent technology for processing the information of the Web. The Semantic Web consists of several hierarchical layers, where the Ontology layer, in form of the OWL Web Ontology Language [37,24] (recommended by the W3C), is currently the highest layer of sufficient maturity. OWL consists of three increasingly expressive sublanguages, namely OWL Lite, OWL DL, and OWL Full. OWL Lite and OWL DL are essentially expressive description logics with an RDF syntax [24]. Ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) [23].

The main contributions of this paper can be summarized as follows:

- We introduce conditional preference bases, which consist of a description logic knowledge base and a finite set of conditional preferences. They are syntactically and semantically inspired by Goldszmidt and Pearl's approach to default reasoning from conditional knowledge bases in System Z^+. We define a notion of consistency for conditional preference bases, and show how consistent conditional preference bases can be used for ranking objects in ontologies.
- We also provide algorithms for computing the rankings relative to a conditional preference base. These algorithms are based on a reduction to deciding whether a description logic knowledge base is satisfiable. More precisely, they require a polynomial number of such satisfiability tests, and thus can all be done in polynomial time whenever the satisfiability tests are possible in polynomial time.

– Finally, we describe an application of this approach in literature search. Search query languages of current search engines are very restricted in their expressive power. There are scientific search engines on the web, however, that have valuable metadata about research publications, authors, organizations, and scientific events. We show that conditional preference bases allow for a more powerful query language, which can exploit this metadata better than the current approaches do. In particular, we give some sample queries that (i) explicitly follow different search strategies, (ii) influence the ranking of the query results, (iii) express quality measures, (iv) cluster query results, or (v) restrict queries to different result types.

2 The Description Logics $\mathcal{SHIF}(\mathrm{D})$ and $\mathcal{SHOIN}(\mathrm{D})$

In this section, we recall the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL, respectively [23]. Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between classes of individuals, respectively. Roughly, a description logic knowledge base encodes subset relationships between classes, the membership of individuals to classes, and the membership of pairs of individuals to binary relations between classes.

Syntax. We first describe the syntax of $\mathcal{SHOIN}(\mathbf{D})$. We assume a set of *elementary datatypes* and a set of *data values*. A *datatype* is either an elementary datatype or a set of data values (called *datatype oneOf*). A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each elementary datatype a subset of $\Delta^{\mathbf{D}}$ and to each data value an element of $\Delta^{\mathbf{D}}$. The mapping $\cdot^{\mathbf{D}}$ is extended to all datatypes by $\{v_1, \ldots\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots\}$. Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ be pairwise disjoint finite nonempty sets of *atomic concepts*, *abstract roles*, *datatype roles*, and *individuals*, respectively. We denote by $\mathbf{R}_A^-$ the set of inverses R^- of all $R \in \mathbf{R}_A$.

A *role* is an element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Every $\phi \in \mathbf{A}$ is a concept, and if $o_1, \ldots, o_n \in \mathbf{I}$, then $\{o_1, \ldots, o_n\}$ is a concept (called *oneOf*). If ϕ, ϕ_1, and ϕ_2 are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then also $(\phi_1 \sqcap \phi_2)$, $(\phi_1 \sqcup \phi_2)$, and $\neg\phi$ are concepts (called *conjunction*, *disjunction*, and *negation*, respectively), as well as $\exists R.\phi$, $\forall R.\phi$, $\geq nR$, and $\leq nR$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. If D is a datatype and $U \in \mathbf{R}_D$, then $\exists U.D$, $\forall U.D$, $\geq nU$, and $\leq nU$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. We write $\top$ and $\bot$ to abbreviate the concepts $\phi \sqcup \neg\phi$ and $\phi \sqcap \neg\phi$, respectively, and we eliminate parentheses as usual.

An *axiom* has one of the following forms: (1) $\phi \sqsubseteq \psi$ (called *concept inclusion axiom*), where ϕ and ψ are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $\phi(a)$ (called *concept membership axiom*), where ϕ is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$) (called *role membership axiom*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (*equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. A *knowledge base* KB is a finite set of axioms. For decidability, number restrictions in KB are restricted to simple abstract roles [25].

The syntax of $\mathcal{SHIF}(\mathbf{D})$ is as the above syntax of $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 2.1. An online store (such as *amazon.com*) may use a description logic knowledge base to classify and characterize its products. For example, suppose that (1) textbooks are books, (2) personal computers and laptops are mutually exclusive electronic products, (3) books and electronic products are mutually exclusive products, (4) any objects on offer are products, (5) every product has at least one related product, (6) only products are related to each other, (7) tb_ai and tb_lp are textbooks, which are related to each other, (8) pc_ibm and pc_hp are personal computers, which are related to each other, and (9) ibm and hp are providers for pc_ibm and pc_hp, respectively. These relationships are expressed by the following description logic knowledge base KB_1:

$$\begin{aligned}
&\mathit{Textbook} \sqsubseteq \mathit{Book};\ \mathit{PC} \sqcup \mathit{Laptop} \sqsubseteq \mathit{Electronics};\ \mathit{PC} \sqsubseteq \neg \mathit{Laptop};\\
&\mathit{Book} \sqcup \mathit{Electronics} \sqsubseteq \mathit{Product};\ \mathit{Book} \sqsubseteq \neg \mathit{Electronics};\ \mathit{Offer} \sqsubseteq \mathit{Product};\\
&\mathit{Product} \sqsubseteq \geq 1\, \mathit{related};\ \geq 1\, \mathit{related} \sqcup \geq 1\, \mathit{related}^{-} \sqsubseteq \mathit{Product};\\
&\mathit{Textbook}(tb_ai);\ \mathit{Textbook}(tb_lp);\ \mathit{PC}(pc_ibm);\ \mathit{PC}(pc_hp);\\
&\mathit{related}(tb_ai, tb_lp);\ \mathit{related}(pc_ibm, pc_hp);\\
&\mathit{provides}(ibm, pc_ibm);\ \mathit{provides}(hp, pc_hp).
\end{aligned}$$

Semantics. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ w.r.t. a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (*abstract*) *domain* $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $\phi \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$. We extend $\cdot^{\mathcal{I}}$ to all concepts and roles, and we define the *satisfaction* of a description logic axiom F in an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, denoted $\mathcal{I} \models F$, as usual [23]. The interpretation $\mathcal{I}$ *satisfies* the axiom F, or $\mathcal{I}$ is a *model* of F, iff $\mathcal{I} \models F$. The interpretation $\mathcal{I}$ *satisfies* a knowledge base KB, or $\mathcal{I}$ is a *model* of KB, denoted $\mathcal{I} \models KB$, iff $\mathcal{I} \models F$ for all $F \in KB$. We say that KB is *satisfiable* (resp., *unsatisfiable*) iff KB has a (resp., no) model. An axiom F is a *logical consequence* of KB, denoted $KB \models F$, iff every model of KB satisfies F.

3 Conditional Preference Bases

In this section, we first define the syntax of conditional preferences, which are intuitively statements of form "generally, if ϕ, then α is preferred over $\neg\alpha$ with strength s". We then define the semantics of such statements in terms of object rankings, taking inspiration from default reasoning from conditional knowledge bases in System Z^+.

Syntax. We assume a finite set of *classification concepts* $\mathcal{C}$ (which are the relevant description logic concepts for defining preference relationships). A *conditional preference* is of the form $(\alpha|\phi)[s]$ with concepts $\phi \in \mathcal{C}$ (called its *body*) and $\alpha \in \mathcal{C}$ (called its *head*), and an integer $s \in \{0, \ldots, 100\}$ (called its *strength*). Informally, $(\alpha|\phi)[s]$ expresses that (i) generally, among the objects satisfying ϕ, the ones satisfying α are preferred over those satisfying $\neg\alpha$, and (ii) this preference relationship holds with strength s. Conditional preferences of the form $(\alpha|\top)[s]$ are also abbreviated as $(\alpha)[s]$. A *conditional*

preference base is a pair $PB = (T, A, P)$, where T is a description logic knowledge base, A is a finite set of concepts from $\mathcal{C}$, and P is a finite set of conditional preferences. Informally, T contains terminological knowledge, and A contains assertional knowledge about an individual o (that is, A actually represents the set of all $C(o)$ such that $C \in A$), while P contains conditional preferences about the individual o (that is, P actually represents the set of all $(\alpha(o)|\phi(o))[s]$ such that $(\alpha|\phi)[s] \in P$). Observe also that the statements in T and A are *strict* (that is, they must always hold), while the ones in P are *defeasible* (that is, they may have exceptions and thus do not always hold), since P may not always be satisfiable as a whole.

Example 3.1. The assertional knowledge "either a PC or a laptop" and the preference relationships "generally, PC's are preferred over laptops with strength 20", "generally, laptops on offer are preferred over PC's on offer with strength 70", and "generally, inexpensive objects are preferred over expensive ones with strength 90" can be expressed by the conditional preference base $PB = (T, A, P)$, where T is the description logic knowledge base from Example 2.1, $A = \{\top \sqsubseteq PC \sqcup \mathit{Laptop}\}$, and $P = \{(PC)[20], (\mathit{Laptop}|\mathit{Offer})[70], (\mathit{Inexpensive})[90]\}$.

Semantics. We now define some basic semantic notions, including objects and object rankings (which are certain functions that map every object to a rank from $\{0, 1, \ldots\} \cup \{\infty\}$), and we then associate with every conditional preference base a set of object rankings as a formal semantics. An *object* o is a set of concepts from $\mathcal{C}$. We denote by $\mathcal{O}_\mathcal{C}$ the set of all objects relative to $\mathcal{C}$. An object o *satisfies* a description logic knowledge base T, denoted $o \models T$, iff $T \cup \{\phi(i) \mid \phi \in o\}$ is satisfiable and entails (resp., does not entail) every concept membership $\phi(i)$ such that $\phi \in o$ (resp., $\phi \notin o$), where i is a new individual. Informally, every object o represents an individual i that is fully specified on $\mathcal{C}$ in the sense that o belongs (resp., does not belong) to every concept $\phi \in o$ (resp., $\phi \notin o$). An object o satisfies a concept $\phi \in \mathcal{C}$, denoted $o \models \phi$, iff $\phi \in o$. An object o satisfies a set of concepts $A \subseteq \mathcal{C}$, denoted $o \models A$, iff o satisfies all $\phi \in A$. A concept ϕ is *satisfiable* iff there exists an object $o \in \mathcal{O}_\mathcal{C}$ that satisfies ϕ. An object o *satisfies* a conditional preference $(\alpha|\phi)[s]$, denoted $o \models (\alpha|\phi)[s]$, iff $o \models \neg\phi \sqcup \alpha$. We say o *satisfies* a set of conditional preferences P, denoted $o \models P$, iff o satisfies all $p \in P$. We say o *verifies* $(\alpha|\phi)[s]$ iff $o \models \phi \sqcap \alpha$. We say o *falsifies* $(\alpha|\phi)[s]$, denoted $o \not\models (\alpha|\phi)[s]$, iff $o \models \phi \sqcap \neg\alpha$. A set of conditional preferences P *tolerates* a conditional preference p *under* a description logic knowledge base T and a set of classification concepts $A \subseteq \mathcal{C}$ iff an object o exists that satisfies $T \cup A \cup P$ (that is, o satisfies T, A, and P) and verifies p. We say P is *under* T *and* A *in conflict* with p iff P does not tolerate p under T and A.

An *object ranking* κ is a mapping $\kappa\colon \mathcal{O}_\mathcal{C} \to \{0, 1, \ldots\} \cup \{\infty\}$ such that $\kappa(o) = 0$ for at least one object $o \in \mathcal{O}_\mathcal{C}$. It is extended to all concepts ϕ as follows. If ϕ is satisfiable, then $\kappa(\phi) = \min\{\kappa(o) \mid o \in \mathcal{O}_\mathcal{C}, o \models \phi\}$; otherwise, $\kappa(\phi) = \infty$. We say κ is *admissible* with a description logic knowledge base T (resp., a set of concepts A) iff $\kappa(o) = \infty$ for all $o \in \mathcal{O}_\mathcal{C}$ such that $o \not\models T$ (resp., $o \not\models A$). We say κ is *admissible* with a conditional preference $(\alpha|\phi)[s]$ iff either $\kappa(\phi) = \infty$ or $\kappa(\phi \sqcap \alpha) + s < \kappa(\phi \sqcap \neg\alpha)$. We say κ is *admissible* with $PB = (T, A, P)$ iff κ is admissible with T, A, and all $p \in P$.

Consistency. The notion of consistency is inspired by the notion of ε-consistency for conditional knowledge bases [1,21]. A conditional preference base PB is *consistent*

(resp., *inconsistent*) iff an (resp., no) object ranking κ exists that is admissible with PB. Notice that $PB = (T, A, P)$ with $P = \emptyset$ is consistent iff $T \cup A$ is satisfiable. We now summarize some results that carry over from conditional knowledge bases.

The following result shows that the existence of an object ranking that is admissible with $PB = (T, A, P)$, where $P \neq \emptyset$, is equivalent to the existence of a preference ranking on P that is admissible with PB. Here, a *preference ranking* σ on a set of conditional preferences P maps each $p \in P$ to an integer. We say that a preference ranking σ on P is *admissible* with $PB = (T, A, P)$ iff every $P' \subseteq P$ that is under T and A in conflict with some $p \in P$ contains some p' such that $\sigma(p') < \sigma(p)$.

Theorem 3.1. *A conditional preference base $PB = (T, A, P)$ with $P \neq \emptyset$ is consistent iff there exists a preference ranking σ on P that is admissible with PB.*

The next result shows that the consistency of PB is equivalent to the existence of an ordered partition of P with certain properties.

Theorem 3.2. *A conditional preference base $PB = (T, A, P)$ with $P \neq \emptyset$ is consistent iff there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that either (a) every P_i, $0 \leq i \leq k$, is the set of all $p \in \bigcup_{j=i}^{k} P_j$ tolerated under T and A by $\bigcup_{j=i}^{k} P_j$, or (b) for every i, $0 \leq i \leq k$, each $p \in P_i$ is tolerated under T and A by $\bigcup_{j=i}^{k} P_j$.*

We call the unique partition in (a) the *z-partition* of PB.

Example 3.2. The conditional preference base PB of Example 2.1 is consistent. Its z-partition is $(P_0, P_1) = (\{(PC)[20], (Inexpensive)[90]\}, \{(Laptop|Offer)[70]\})$.

4 Ranking Objects under Conditional Preference Bases

In this section, we define object rankings that reflect the conditional preferences encoded in a consistent conditional preference base $PB = (T, A, P)$.

We first rewrite P from a set of defeasible statements to a set of strict statements $P^\star$. Intuitively, this is done by adding exceptions to the bodies of conditional preferences.

Example 4.1. Let the conditional preference base $PB = (T, A, P)$ be given by T and A as in Example 3.1 and $P = \{(PC)[20], (Laptop|Offer)[70]\}$. Ignoring the strengths, P encodes that "PCs are preferred over laptops, as long as they are not on offer, because in that case, laptops are preferred over PCs". That is, for technical reasons, laptops on offer always falsify the conditional preference $p = (PC)[20]$. When computing the rank of laptops on offer, we have to avoid such falsifications. We do this by rewriting p and thus PB. The rewritten conditional preference base $PB^\star = (T, A, P^\star)$ is given by $P^\star = \{(PC|\neg Offer)[20], (Laptop|Offer)[70]\}$. It is obtained from PB by adding the exception $\neg Offer$ to the body of $(PC)[20]$.

A conditional preference base $PB = (T, A, P)$ is *flat* iff its z-partition is given by (P) and thus consists only of one component. Algorithm *flatten* in a companion paper [30] transforms a consistent conditional preference base $PB = (T, A, P)$ into an equivalent flat conditional preference base, denoted $PB^\star = (T, A, P^\star)$.

Table 1. The object rankings κ^{sum} and κ^{lex}

	PC	*Laptop*	*Offer*	κ^{sum}	κ^{lex}
o_1	**false**	**false**	**false**	∞	∞
o_2	**false**	**false**	**true**	∞	∞
o_3	**false**	**true**	**false**	21	1
o_4	**false**	**true**	**true**	0	0

	PC	*Laptop*	*Offer*	κ^{sum}	κ^{lex}
o_5	**true**	**false**	**false**	0	0
o_6	**true**	**false**	**true**	71	2
o_7	**true**	**true**	**false**	∞	∞
o_8	**true**	**true**	**true**	∞	∞

We are now ready to define the object rankings κ^{sum} and κ^{lex}. Informally, κ^{sum} associates with every object (as a penalty) the sum of the strengths of all conditional preferences in $P^\star$ that are falsified by o. Roughly, objects with smaller values under κ^{sum} are those that satisfy more conditional preferences with larger strengths. Formally, κ^{sum} is defined as follows for all objects $o \in \mathcal{O}_\mathcal{C}$:

$$\kappa^{sum}(o) = \begin{cases} \infty & \text{if } o \not\models T \cup A \\ \sum\limits_{p=(\alpha|\phi)[s] \in P^\star \,:\, o \not\models p} s+1 & \text{otherwise.} \end{cases} \tag{1}$$

The object ranking κ^{lex}, in contrast, is based on a lexicographic order. Roughly, objects with smaller values under κ^{lex} are those that satisfy more conditional preferences with larger strengths, where satisfying one conditional preference of strength s is strictly preferred to satisfying any set of conditional preferences of strength at most $s-1$. Formally, κ^{lex} is defined as follows for all objects $o \in \mathcal{O}_\mathcal{C}$ (where n_j with $j \in \{0, \ldots, 100\}$ is the number of all $p \in P^\star$ of strength j):

$$\kappa^{lex}(o) = \begin{cases} \infty & \text{if } o \not\models T \cup A \\ \sum\limits_{i=0}^{100} |\{p = (\alpha|\phi)[i] \in P^\star \mid o \not\models p\}| \cdot \Pi_{j=0}^{i-1}(n_j+1) & \text{otherwise.} \end{cases} \tag{2}$$

Example 4.2. The object rankings κ^{sum} and κ^{lex} for PB of Example 4.1 are shown in Fig. 1. For example, under both κ^{sum} and κ^{lex}, the object o_4 is strictly preferred over o_3, as desired, since $\kappa^{sum}(o_4) < \kappa^{sum}(o_3)$ and $\kappa^{lex}(o_4) < \kappa^{lex}(o_3)$, respectively.

Summarizing, every object ranking $\kappa \in \{\kappa^{sum}, \kappa^{lex}\}$ of a conditional preference base PB represents the preference relationships encoded in PB. For every (fully specified) object o, the *rank* of o under PB is given by $\kappa(o)$. Every object ranking $\kappa \in \{\kappa^{sum}, \kappa^{lex}\}$ can also be used to compare two objects $o, o' \in \mathcal{O}_\mathcal{C}$ as follows. The *distance* between o and o' under PB is defined as $|\kappa(o) - \kappa(o')|$. Furthermore, the (*credulous*) *rank* of a partially specified object (which is simply a concept) ϕ under PB is defined as $\min_{o \in \mathcal{O}_\mathcal{C} : o \models \phi} \kappa(o)$. Finally, the (*credulous*) *distance* between two partially specified objects ϕ and ϕ' is defined as $\min_{o, o' \in \mathcal{O}_\mathcal{C} : o \models \phi, o' \models \phi'} |\kappa(o) - \kappa(o')|$.

5 Algorithms and Complexity

There are several computational tasks related to conditional preference bases $PB = (T, A, P)$. First, deciding the consistency of PB is done by algorithm *consistency* in

Algorithm *sum-ranking*

Input: conditional preference base $PB = (T, A, P)$ and set of objects $\mathcal{O} \subseteq \mathcal{O}_{\mathcal{C}}$.
Output: ranking κ^{sum} on $\mathcal{O}$ for PB, if PB is consistent; *nil*, otherwise.

1. **if** PB is inconsistent **then return** *nil*;
2. **for each** $o \in \mathcal{O}$ **do if** $o \models A \cup T$ **then** $\kappa(o) := 0$ **else** $\kappa(o) := \infty$;
3. **if** $P = \emptyset$ **then return** κ;
4. $(T, A, P) :=$ *flatten*(T, A, P);
5. **for each** $o \in \mathcal{O}$ such that $\kappa(o) \neq \infty$ **do**
6. **for each** $p = (\alpha|\phi)[s] \in P$ **do if** $o \not\models p$ **then** $\kappa(o) := \kappa(o) + s + 1$;
7. **return** κ.

Fig. 1. Algorithm *sum-ranking*

Algorithm *lex-ranking*

Input: conditional preference base $PB = (T, A, P)$ and set of objects $\mathcal{O} \subseteq \mathcal{O}_{\mathcal{C}}$.
Output: ranking κ^{lex} on $\mathcal{O}$ for PB, if PB is consistent; *nil*, otherwise.

1. **if** PB is inconsistent **then return** *nil*;
2. **for each** $o \in \mathcal{O}$ **do if** $o \models A \cup T$ **then** $\kappa(o) := 0$ **else** $\kappa(o) := \infty$;
3. **if** $P = \emptyset$ **then return** κ;
4. $(T, A, P) :=$ *flatten*(T, A, P);
5. **for each** $o \in \mathcal{O}$ such that $\kappa(o) \neq \infty$ **do begin**
6. $n := 1$;
7. **for each** $i := 0$ **to** 100 **do begin**
8. $h := 0$;
9. **for each** $p = (\alpha|\phi)[i] \in P$ **do if** $o \not\models p$ **then** $h := h + i + 1$
10. $\kappa(o) := \kappa(o) + h \cdot n$;
11. $n := n \cdot (|\{(\alpha|\phi)[s] \in P \mid s = i\}| + 1)$
12. **end**
13. **end**;
14. **return** κ.

Fig. 2. Algorithm *lex-ranking*

a companion paper [30] (which returns the z-partition of PB, if PB is consistent, and *nil*, otherwise), which generalizes an algorithm for deciding ε-consistency in default reasoning [21]. The extended algorithm is essentially based on $O(|P|^2)$ tests whether a description logic knowledge base is satisfiable. Second, rewriting PB to an equivalent flat conditional preference base $PB^\star$ is done by algorithm *flatten* in a companion paper [30], which is similar to a rewriting algorithm in fuzzy default reasoning [16], and which requires $O(|P|^2)$ description logic satisfiability tests. Finally, computing the ranking functions κ^{sum} and κ^{lex} is done by algorithms *sum-* and *lex-ranking* in Figs. 1 and 2, respectively, in a polynomial number of description logic satisfiability tests.

Theorem 5.1. *Given a conditional preference base $PB = (T, A, P)$ and a set of objects $\mathcal{O} \subseteq \mathcal{O}_{\mathcal{C}}$, computing the rankings κ^{sum} and κ^{lex} on $\mathcal{O}$ relative to PB can be done in $O(|P| \cdot (|P| + |\mathcal{C}| \cdot |\mathcal{O}|))$ description logic satisfiability tests.*

Hence, under the assumption that $|P|$ and $|\mathcal{C}|$ are bounded by a constant (which is a reasonable assumption in the application in literature search below), computing the rankings κ^{sum} and κ^{lex} can be done in $O(|\mathcal{O}|)$ description logic satisfiability tests.

Furthermore, if we restrict the class of description logic expressions in PB in such a way that the above satisfiability tests on description logic knowledge bases can be done in polynomial time (for example, such as in DL-Lite [12]), then all the described computational tasks can also be solved in polynomial time.

6 Application: Literature Search

In this section, we describe an application in literature search for the above approach to ranking objects relative to a conditional preference base.

Background. A very important and time consuming task of researchers is finding publications. There exist a lot of possibilities to find relevant research publications over the internet. For instance, there are portals for research publications, portals for ejournals, special purpose search engines for researchers (for example, CiteSeer and Google Scholar), specialized databases, publication databases of institutions, and bibliographic online catalogues. It seems that there is a trend to more diversity and quality regarding online search engines. On the other hand, the "tremendous increase in the quantity and diversity of easily available research publications has exacerbated the problems of information overload for researchers attempting to keep abreast of new relevant research, especially in rapidly advancing fields" [7].

A very powerful instrument for search engines are citation indexes, which can be very well exploited for search processes. Garfield [19] examined the possibilities and advantages of citation indexes. There are a lot of advantages of citation indexes compared to traditional subject indexes. The quality of references tends to be higher than the quality of title words and keywords. Using citation indexes enhances the search productivity (finding the largest possible number of relevant publications) and the search efficiency (minimizing the number of irrelevant publications). Citations are semantically more stable than keywords. Citation indexes can be used in many ways, for example, finding relevant publications through backward and forward navigation, finding out the importance of publications, and identifying research trends [13].

CiteSeer and Google Scholar have recognized the work of Garfield and are using the valuable information of citations. In Google, for example, the ranking algorithm PageRank is based on the linking of web resources [10]. CiteSeer automatically detects scientific publications on the Web and extracts the necessary metadata (citations, citation context, title, etc.), builds the citation index, and performs the full-text indexing [27]. For the ranking of query results, CiteSeer has adopted the ranking algorithm of Google. The CiteSeer database is queried by simple keyword search and returns a list of indexed publications [20]. For each publication, CiteSeer offers a query-sensitive summary, containing the citation context of the publication, links to similar documents and links to author homepages [27]. The user follows citations by browsing the links. For each query result, there are a lot of pieces of information and links that can be used to browse the database. In Google Scholar, one also uses keyword searches that can be

restricted, for example, to authors or titles. The search result contains a list of publications that match the query. For each publication, there is a link "cited by" that leads to a list of publications citing the discovered publication.

There are a lot of good search strategies that a researcher can use for the task of finding relevant scientific publications. Bates [4] has identified the following six important information search strategies:

- *Footnote chasing:* Following up footnotes (that is, references) found in publications. This can be done in successive leaps.
- *Citation searching:* Looking for publications that cite certain publications.
- *Journal run:* Identification of a central journal in a research area and then looking up publications in relevant volumes.
- *Area scanning:* Browsing resources that are physically collocated with resources that are regarded as relevant. A good example is a book shelf in a library. In a digital library, one could exploit the classification of resources.
- *Subject searches:* The usage of subject descriptors such as keywords to find relevant publications.
- *Author searching:* To find other publications of an author, which may have a similar topic as a publication one already knows of.

The first two strategies are supported by citation indexes. It would be helpful if the above mentioned search strategies could be explicitly supported by the query languages of search engines. The power of search query languages of scientific search engines should go beyond the classical Boolean keyword searching. It should be enhanced to ask more elaborated queries. The search strategies citation searching, subject searches, and author searching are well supported by CiteSeer and Google Scholar. The search strategy footnote chasing is well supported by CiteSeer, although normally not all citations to a publication are listed. There is no direct support for the search strategy footnote chasing with Google Scholar. The publication has to be found and downloaded. Then the references have to be looked up in the document. CiteSeer and Scholar Google are no help for the search strategy journal run. In order to use the search strategy journal run, one could use Google to find the web site of the journal and then one can browse through the journal's volumes. The search strategy area scanning is not supported by CiteSeer, Google Scholar, and Google.

Actually, to use all the search strategies footnote chasing, citation searching, journal run, subject searches, and author searching, one has to use all the above search engines, and there is no way to exploit the search strategies by the formulation of the search queries. What is also not supported by the mentioned search engines is the possibility to exploit relationships like citations or co-authorship by the formulation of queries. To date, search query languages of most web search engines have little expressive power for formulating semantic queries, cannot be used to explicitly influence the ranking of query results, have no possibilities to formulate ones own quality measures for the query results, have no possibilities to restrict the query results to certain result types (for example, authors, journals, conferences, keywords, and publications), and have no possibilities to influence the clustering of query results. Of course, even most scientists normally do not want to learn a complex query language. Therefore, one has to think

about good query assistants that help formulating sophisticated queries. Nevertheless, when the benefit of more sophisticated queries becomes clear, we are convinced that researchers will use such query languages instead of the query assistants.

Literature Search via Conditional Preference Bases. In this section, we show that our approach to conditional preferences bases allows for expressing more sophisticated search queries, and avoids the above mentioned deficiencies. The presented examples also show the expressive power of the formalism proposed in this paper.

The strict terminological knowledge is informally described as follows. We assume the concepts *Publication*, *JournalPublication*, *ConfPublication*, *Person*, *Publicationmedium*, *Journal*, *Proceedings*, *Keyword*, *Event*, *Conference*, and *Workshop*, which are related by the concept inclusion axioms *JournalPublication* $\sqsubseteq$ *Publication*, *ConfPublication* $\sqsubseteq$ *Publication*, *Conference* $\sqsubseteq$ *Event*, *Workshop* $\sqsubseteq$ *Event*, *Journal* $\sqsubseteq$ *Publicationmedium*, and *Proceedings* $\sqsubseteq$ *Publicationmedium*. We assume the roles *Author* (relating *Publication* and *Person*), *Coauthor* (on *Person*), *Cite* (on *Publication*), *Publishedin* (relating *Publication* and *Publicationmedium*), *Keywords* (relating *Publication* and *Keyword*), and *hasPublicationmedium* (relating *Event* and *Publicationmedium*). Moreover, the concept *Publication* has the attributes *year*, *title*, *publishedat*, and *type*. Finally, we assume the unary function *in_title* of type *string*$\rightarrow$ *Publication*.

In the following, some literature search queries are associated with a corresponding conditional preference base $PB = (T, A, P)$, expressed as the conjunction of all the elements in $A \cup P$. For example, consider the following query Q (which supports the search strategy *subject searches*): We are looking for papers with the word "matching" in the title. In case of a conference paper, we prefer papers of international conferences to papers of national conferences. This query is expressed by the following conjunction:

$$C = \mathit{Publication} \sqcap \mathit{in\_title}(\text{“matching”}) \sqcap \\ (\mathit{type}(\text{“international”})|\mathit{ConfPublication})[70] \sqcap (\mathit{ConfPublication})[80]\,,$$

which in turn stands for the conditional preference base $PB = (T, A, P)$, where $T = \emptyset$,

$$A = \{\mathit{Publication},\ \mathit{in\_title}(\text{“matching”})\}, \text{ and}$$
$$P = \{(\mathit{type}(\text{“international”})|\mathit{ConfPublication})[70],\ (\mathit{ConfPublication})[80]\}\,.$$

Query Q contains two conditional preferences (with the two strengths 70 and 80, respectively, which are directly specified by the user). Intuitively, an object that fulfills query Q has to be a publication with the word "matching" in the title and it should possibly satisfy the two conditional preferences. Publications that satisfy the conditional preferences have a lower rank than publications that falsify them. Query Q therefore divides the publications in the query result into three groups as follows: first international conference publications (lowest rank), second national conference publications (second lowest rank), and third non-conference publications (highest rank).

We now provide several other queries, expressed in the textual and the conjunctive way. The following queries support the search strategy *footnote chasing*:

(1) All references that Ian Horrocks cited in his papers:
$\exists \mathit{Cite}^{-}.\exists \mathit{Author}.\{\text{“Ian Horrocks”}\}$.
(2) Journal publications that were cited by Ian Horrocks:
$\mathit{JournalPublication} \sqcap \exists \mathit{Cite}^{-}.\exists \mathit{Author}.\{\text{“Ian Horrocks”}\}$.

The following queries support the search strategy *citation searching*:

(3) All publications that cite the paper "Weaving the Web" of Tim Berners Lee:
$\exists Cite.(title(\text{"Weaving the Web"}) \sqcap \exists Author.\{\text{"Tim Berners Lee"}\})$.

(4) All publications that cite publications of Ian Horrocks:
$\exists Cite.\exists Author.\{\text{"Ian Horrocks"}\}$.

(5) All publications that cite papers of ISWC in the year 2000:
$\exists Cite.(ConfPublication \sqcap publishedat(\text{"ISWC"}) \sqcap =_{2000}(year))$.

The following queries support the search strategy *journal run*:

(6) All publications of ISWC after the year 2001:
$ConfPublication \sqcap publishedat(\text{"ISWC"}) \sqcap \geq_{2001}(year)$.

(7) All journals that were cited by Ian Horrocks:
$Journal \sqcap \exists Publishedin^{-}.(\exists Cite^{-}.\exists Author.\{\text{"Ian Horrocks"}\})$.

(8) All conferences with publications that contain the keywords "elearning" and "Semantic Web":
$Conference \sqcap \exists hasPublicationmedium.\exists Publishedin^{-}$
$(\exists Keywords.\{\text{"elearning"}\} \sqcap \exists Keywords.\{\text{"Semantic Web"}\})$.

The following queries support the search strategy *subject searches*:

(9) All publications with "Semantic Web" in the title that were cited at least five times:
$in\_title(\text{"Semantic Web"}) \sqcap \geq_5 Cite^{-}$.

(10) All publications with "Semantic Web" in the title, that contain at least four literature references that are cited at least ten times:
$in\_title(\text{"Semantic Web"}) \sqcap \geq_4 Cite.(\geq_{10} Cite^{-})$.

(11) All publications with the keyword "Semantic Web" and the keywords "OWL" and "DAML+OIL". The ranking is influenced by the strengths of the keywords:
$\exists Keywords.\{\text{"Semantic Web"}\} \sqcap (\exists Keywords.\{\text{"OWL"}\})[70] \sqcap$
$(\exists Keywords.\{\text{"DAML+OIL"}\})[20]$.

The following queries support the search strategy *author searching*:

(12) All publications of Ian Horrocks:
$\exists Author.\{\text{"Ian Horrocks"}\}$.

(13) All publications of authors who have a joint publication with Tim Berners-Lee:
$\exists Author.\exists Coauthor.\{\text{"Tim Berners-Lee"}\}$.

(14) All publications that cite publications of coauthors of Tim Berners-Lee and himself, giving a higher rank to publications that cite Tim Berners-Lee:
$(\exists Cite.\exists Author.\exists Coauthor.\{\text{"Tim Berners-Lee"}\})[30] \sqcap$
$(\exists Cite.\exists Author.\{\text{"Tim Berners-Lee"}\})[80]$.

Query 14 divides the publications in the query result into three groups as follows: first publications that cite publications that Tim Berners-Lee wrote with colleagues (lowest rank), second publications that cite publications where Tim Berners-Lee was the only author (second lowest rank), and third publications that cite publications of coauthors of Tim Berners-Lee, where Tim Berners-Lee was not an author (highest rank).

Note that queries 9 and 10 include a user-defined quality measure. Query 7 (resp., 8) has the result type journal (resp., conference). Queries 11 and 14 are directly influencing the ranking of the query results, and they are also clustering the query results.

Two important measures for retrieval systems are precision and recall. Let Q be a query, let R be the set of relevant documents to the query Q, let A be the set of documents that a retrieval system returns for Q, and let $Ra = R \cap A$ be the set of relevant documents to Q within A. Then, the notions of *precision* and *recall* are defined by *precision* $= |Ra| \,/\, |A|$ and *recall* $= |Ra| \,/\, |R|$, respectively [3].

Queries 1–7, 12, 13, and 14 are examples where the information need could be specified very precisely, resulting in relative small query results. These are examples that lead to a higher precision and a higher recall. Queries 8–11 contain elements that are based on string comparisons. Although these elements are restricted to titles and keywords, they cannot correctly express the information need. This is a known problem for conventional search engines. Queries 11 and 14 contain conditional preferences. The purpose of the conditional preferences is to influence the clustering and the ranking of the query results. Although there is no effect on precision and recall, this helps the user. Normally users just look at the top results of a result list. The user defined ranking and clustering makes the ranking more transparent to the user and increases the likelihood that the user actually recognizes the most important query results.

7 Related Work

We now give an overview on default reasoning from conditional knowledge bases, and we discuss (less closely) related work on skyline queries and rankings in databases.

The literature contains several different proposals for default reasoning from conditional knowledge bases and extensive work on its desired properties. The core of these properties are the rationality postulates of System P by Kraus et al. [26], which constitute a sound and complete axiom system for several classical model-theoretic entailment relations under uncertainty measures on worlds. They characterize classical model-theoretic entailment under preferential structures, infinitesimal probabilities, possibility measures [15], and world rankings. They also characterize an entailment relation based on conditional objects [14]. A survey of all these relationships is given in [5,18]. Mainly to solve problems with irrelevant information, the notion of rational closure as a more adventurous notion of entailment was introduced by Lehmann [28]. It is in particular equivalent to entailment in System Z by Pearl [32] (which is generalized to variable-strength defaults in System Z^+ by Goldszmidt and Pearl [22]) and to the least specific possibility entailment by Benferhat et al. [5]. Recently, also generalizations of many of the above approaches to probabilistic and fuzzy default reasoning have been proposed (see especially [29] and [16], respectively).

In the database context, the work [9] proposes an extension of database systems by a *Skyline* operation, which filters out a set of interesting points from a potentially large set of data points. It presents an extension of SQL by Skyline queries along with algorithms for them. The work [2] proposes several approaches to rank database query results, while [11] focuses on top-k query evaluation in web databases, and [31] proposes a decentralized top-k query evaluation algorithm for peer-to-peer networks. In contrast to our approach, none of the above works deals with ranking objects relative to conditional preferences of the form "generally, in the context ϕ, property α is preferred over property $\neg\alpha$ with strength s" in the framework of expressive description logics.

8 Summary and Outlook

We have presented an approach to conditional preference bases, which consist of a description logic knowledge base and a finite set of conditional preferences, and which are given a qualitative probabilistic formal semantics in a generalization of Goldszmidt and Pearl's System Z^+. We have defined the notion of consistency for conditional preference bases and shown how consistent conditional preference bases can be used for ranking objects in ontologies. We have also provided algorithms for these rankings.

We have demonstrated the usefulness of the presented approach in the area of literature search. Search query languages of current search engines are very restricted in their expressive power. There are scientific search engines on the web, however, that have valuable metadata about research publications, authors, organizations, and scientific events. We have shown that conditional preference bases allow for a more powerful query language, which can exploit this metadata better than the current approaches do. In particular, we have given some sample queries that (i) explicitly follow different search strategies, (ii) influence the ranking of the query results, (iii) express quality measures, (iv) cluster query results, or (v) restrict queries to different result types.

An interesting topic of future research is to explore further applications of the presented approach, for example, in personalization tasks and recommender systems.

Acknowledgments. This work has been partially supported by a Heisenberg Professorship of the German Research Foundation (DFG). We thank the reviewers for their constructive comments, which helped to improve this work.

References

1. E. W. Adams. *The Logic of Conditionals*, volume 86 of *Synthese Library*. D. Reidel, Dordrecht, Netherlands, 1975.
2. S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *Proceedings CIDR-2003*.
3. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
4. M. Bates. The design of browsing and berrypicking techniques for the on-line search interface. *Online Review*, 13(5):407–431, 1989.
5. S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artif. Intell.*, 92(1–2):259–276, 1997.
6. T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, 1999.
7. K. D. Bollacker, S. Lawrence, and C. L. Giles. Discovering relevant scientific literature on the web. *IEEE Intelligent Systems*, 15(2):42–47, 2000.
8. C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.
9. S. Börzsönyi, D. Kossmann, and K. Stocker. The Skyline operator. In *Proceedings ICDE-2001*, pp. 421–430.
10. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings WWW-1998*, pp. 107–117.
11. N. Bruno, L. Gravano, and A. Marian. Evaluating top-k queries over web-accessible databases. In *Proceedings ICDE-2002*, pp. 369–382.

12. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proceedings AAAI-2005*, pp. 602–607.
13. C. Ding, C.-H. Chi, J. Deng, and C.-L. Dong. Citation retrieval in digital libraries. In *Proceedings IEEE SMC-1999*, volume 2, pp. 105–109.
14. D. Dubois and H. Prade. Conditional objects as nonmonotonic consequence relationships. *IEEE Trans. Syst. Man Cybern.*, 24(12):1724–1740, 1994.
15. D. Dubois and H. Prade. Possibilistic logic, preferential models, non-monotonicity and related issues. In *Proceedings IJCAI-1991*, pp. 419–424.
16. F. D. de Saint-Cyr and H. Prade. Handling uncertainty in a non-monotonic setting, with applications to persistence modeling and fuzzy default reasoning. In *Proceedings KR-2006*.
17. D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.
18. D. M. Gabbay and P. Smets, editors. *Handbook on Defeasible Reasoning and Uncertainty Management Systems*. Kluwer Academic, Dordrecht, Netherlands, 1998.
19. E. Garfield. *Citation Indexing — Its Theory and Application in Science, Technology, and Humanities*. John Wiley & Sons, New York, 1979.
20. C. L. Giles, K. D. Bollacker, and S. Lawrence. CiteSeer: An automatic citation indexing system. In *Proceedings ACM Digital Libraries 1998*, pp. 89–98, 1998.
21. M. Goldszmidt and J. Pearl. On the consistency of defeasible databases. *Artif. Intell.*, 52(2):121–149, 1991.
22. M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artif. Intell.*, 84(1–2):57–112, 1996.
23. I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *J. Web Semantics*, 1(4):345–357, 2004.
24. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. Web Semantics*, 1(1):7–26, 2003.
25. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings LPAR-1999*, pp. 161–180, *LNCS* 1705, Springer.
26. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 14(1):167–207, 1990.
27. S. Lawrence, K. D. Bollacker, and C. L. Giles. Indexing and retrieval of scientific literature. In *Proceedings CIKM-1999*, pp. 139-146.
28. D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artif. Intell.*, 55(1):1–60, 1992.
29. T. Lukasiewicz. Weak nonmonotonic probabilistic logics. *Artif. Intell.*, 168: 119–161, 2005.
30. T. Lukasiewicz and J. Schellhase. Variable-strength conditional preferences for matchmaking in description logics. In *Proceedings KR-2006*.
31. W. Nejdl, W. Siberski, U. Thaden, and W.-T. Balke. Top-k query evaluation for schema-based peer-to-peer networks. In *Proceedings ISWC-2004*, pp. 137–151.
32. J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. In *Proceedings TARK-1990*, pp. 121–135.
33. D. Poole and C. Smyth. Type uncertainty in ontologically-grounded qualitative probabilistic matching. In *Proceedings ECSQARU-2005*, pp. 763–774.
34. C. Smyth and D. Poole. Qualitative probabilistic matching with hierarchical descriptions. In *Proceedings KR-2004*, pp. 479–487.
35. M. Stempfhuber. Keep the best — Forget the rest? Towards models for CRISs integrating heterogeneous information. In *Proceedings CRIS-2004*.
36. S.-W. Tan and J. Pearl. Specification and evaluation of preferences under uncertainty. In *Proceedings KR-1994*, pp. 530–539.
37. W3C. OWL web ontology language overview, 2004. W3C Recommendation (10 February 2004). Available at `www.w3.org/TR/2004/REC-owl-features-20040210/`.