A Rank Based Description Language for Qualitative Preferences

Gerhard Brewka
University of Leipzig
Computer Science Department
Leipzig, Germany
email: brewka@informatik.uni-leipzig.de

Abstract

In this paper we develop a language for representing complex qualitative preferences among problem solutions. We use ranked knowledge bases to represent prioritized goals. A basic preference description, that is a ranked knowledge base together with a preference strategy, defines a preference relation on models which represent problem solutions. Our language allows us to express nested combinations of preference descriptions using various connectives. This gives the user the possibility to represent her preferences in a natural, concise and flexible manner.[1]

Introduction

In this paper we develop a language for specifying complex, qualitative preferences among potential problem solutions. Preferences play a crucial role in many areas of AI: in soft constraint solving constraints may have different priority, in decision making or planning some goals may be more important than others, in configuration some properties of the system to be designed are more critical than others, and so on.

By a solution we mean an assignment of a certain value d to each variable v in given set of variables V such that d is taken from the finite domain of v. Without loss of generality, we will restrict our discussion here to the boolean case where the values for each variable are $true$ or $false$. Solutions thus correspond to interpretations in the sense of classical propositional logic. Moreover, we also assume that background knowledge may be given in the form of a set of propositional formulas B. This background knowledge further constrains the set of interpretations: only models of B are considered as potential solutions. We are thus looking for ways of specifying preferences among such models in a concise yet flexible way.

The number of models is exponential in the number of variables. For this reason it is, in general, impossible for a user to describe her preferences by enumerating all pairs of the preference relation among models. This is where logic comes into play.

Traditionally, logic is used for proving theorems. Here, we are not so much interested in logical consequence, we are interested in whether a model satisfies a formula or not. In the simplest case we can use a single formula f, interpret it as a goal, and say a model m_1 is preferred to model m_2 (denoted $m_1 > m_2$) iff $m_1 \models f$ and $m_2 \not\models f$. Note that already in this simple case the formula f alone does not tell us anything about the preference relation. We need additionally a specification of how to use it for determining preferences among models. In the example, f might as well be a condition the user wants to avoid. In other words, we may have $m_1 > m_2$ iff $m_1 \not\models f$ and $m_2 \models f$. Only if the specification of how the formula is to be used is given, a preference order on models can be derived.

In the general case, a single formula is not sufficient and we need a set of formulas F rather than a single formula. We obviously may have more than one goal. Since it may well be impossible to satisfy all of them, a preference relation among the elements of F is useful to distinguish important from less important goals. To express the preferences among goals we will use ranked knowledge bases (RKBs) in this paper (Brewka 1989; Benferhat *et al.* 1993; Pearl 1990; Goldszmidt & Pearl 1991) which are sometimes also called stratified knowledge bases. Such knowledge bases have proven fruitful in a number of approaches. A brief introduction will be given in the next section. Intuitively, the rank $rank(f)$ of a formula f in an RKB is an integer expressing its relative importance. Again, an RKB alone is not sufficient to determine the preference relation on models. We need in addition a recipe of how to use the RKB for this purpose, in other words, we need a strategy.

Although the use of integers is convenient here, RKBs are often used in a purely qualitative way where the actual numbers are irrelevant. What counts is only the total preorder $\geq$ on formulas represented through the integers, where $f_1 \geq f_2$ iff $rank(f_1) \geq rank(f_2)$.

Our focus in this paper will be entirely on these qualitative approaches. This excludes, for instance, approaches which consider ranks as rewards and maximize their sum, as is often done in soft constraint satisfaction (Schiex, Fargier, & Verfaillie 1995). For an excellent overview of some of these approaches see (Lang 2004). Numerical approaches certainly are highly interesting. Nevertheless, we believe that they are better treated in the realm of classical decision theory. The strength of RKBs lies in their potential for modeling qualitative preferences.

[1]This paper was accepted for ECAI-04, the European Conference on Artificial Intelligence

We will discuss several qualitative strategies which have been used in combination with an RKB. Different strategies reflect different meanings a user can associate with the importance ranks. Since there is no single best reading of such ranks, there is no single best strategy. We thus think it is important to give users the ability to choose and possibly combine different strategies in flexible ways. Our main contribution is thus a language for defining complex preferences among models. The basic building blocks are pairs consisting of a strategy and an RKB. Our language also allows for (nested) combinations of preference expressions of this kind using different combination methods.

Throughout the paper the RKBs we use contain formulas representing goals or desires. Independently of the chosen strategy, making more formulas true can never decrease the quality of a model. Some authors have also investigated rejections, that is formulas which should be falsified (Benferhat *et al.* 2002). It turns out that the rejection of p can be modeled using the goal $\neg p$, given an adequate strategy. Our choice of a goal based approach thus does not reduce generality.

The rest of the paper is organized as follows. In the next section we give a brief reminder on ranked knowledge bases. The following section introduces basic preference expressions, consisting of an RKB together with one of 4 qualitative strategies. We also investigate their relationship. The subsequent section defines our full preference description language. In this language, expressions can be combined using various operators. We then illustrate our language using a movie selection example. The last section discusses related work and concludes.

Ranked Knowledge Bases

A ranked knowledge base (RKB), sometimes also called stratified knowledge base, is a set F of propositional formulas together with a total preorder $\geq$ on F. A preorder is a transitive and reflexive relation, totality means that for each $f_1, f_2 \in F$ we have $f_1 \geq f_2$ or $f_2 \geq f_1$. Usually, RKBs are represented in one of the following ways:

1. as a sequence $(F_1, \ldots, F_n)$ of sets of formulas such that $f_1 \geq f_2$ iff for some i, j: $f_1 \in F_i, f_2 \in F_j$ and $i \geq j$.
2. as a set of ranked formulas (f, k), where f is a propositional formula and k, the rank of f, is a non-negative integer such that $f_1 \geq f_2$ iff $rank(f_1) \geq rank(f_2)$.

The two representations of RKBs are clearly equivalent: the rank of a formula corresponds to the set index in the first formulation. For convenience we will mostly use the second one in this paper. Note that starting from a pair $(F, \geq)$ one always gets a set of ranked formulas where each formula has a unique rank.[2]

Intuitively, we consider formulas with higher rank to be more important than those with lower rank.[3] The exact meaning of the ranks depends on the chosen preference strategy.

Different ways of defining consequence relations for RKBs have been defined in the literature. In (Brewka 1989) an inclusion based method was used to define preferred maximal consistent subsets (called preferred subtheories in (Brewka 1989)) of the premises. A maximal subset S_1 is strictly preferred to S_2 iff there is a rank r such that the formulas of rank r in S_1 are a proper superset of those in S_2, and for all ranks higher than r, S_1 and S_2 agree on the contained formulas. Benferhat and colleagues (Benferhat *et al.* 1993) investigated ranked knowledge bases under a cardinality based criterion. To define preferred maximal consistent subsets, they take the number of formulas satisfied in a particular stratum into account. System Z (Pearl 1990; Goldszmidt & Pearl 1991) generates a ranking from a knowledge base of rules which gives more importance to more specific rules. Intuitively, to determine whether a model M is preferred, the lowest rank r is considered for which M satisfies all rules of degree r and higher. A close connection between System Z and possibilistic logic was established in (Benferhat, Dubois, & Prade 2002). The major difference is that possibilistic logic uses reals in the unit interval rather than integers.

In a possibilistic setting, Benferhat and colleagues (Benferhat *et al.* 2002) investigated bipolar preferences based on the maximal degree of a satisfied goal (a model is better the higher the maximal degree) and the maximal degree of a satisfied rejection (a model is the better the smaller the maximal degree).

Since all of these strategies from the literature are of interest, the language to be developed in the next sections will allow the user to pick the one she has in mind when specifying preferences through a ranked knowledge base, and to combine them in a flexible manner.

Basic preference expressions

In this and the following section we define the language LPD for expressing complex preferences among models. We identify 4 basic qualitative strategies which we consider fundamental, given preferences among goals are specified using RKBs. In our language we use identifiers taken from the set

$$Strat = \{\top, \kappa, \subseteq, \#\}.$$

for particular strategies. The meaning of these identifiers will be defined shortly.

Definition 1 *A basic preference description is a pair (s, K) consisting of a basic strategy identifier s and an RKB K.*

Rather than using pair notation $(s, \{(f_1, r_1), \ldots, (f_n, r_n)\})$ or (s, K), we will often use a strategy identifier as an upper index for the RKB, that is, we write $\{(f_1, r_1), \ldots, (f_n, r_n)\}^s$ or K^s, respectively.

A basic preference description defines a preorder $\geq$ (that is, a transitive and reflexive relation) on models. As usual, the preorder implicitly defines an associated strict partial order defined by $m_1 > m_2$ iff $m_1 \geq m_2$ and not $m_2 \geq m_1$.

[2] To represent a set of ranked formulas where a formula f has more than one rank as a pair $(F, \geq)$, one needs syntactic variants of f, that is, equivalent yet syntactically different formulas.

[3] (Brewka 1989) uses the reverse numbering, that is F_1 is the most important set. We find it more intuitive to express higher importance with higher indices.

Let $K = \{(f_i, v_i)\}$ be an RKB, s a basic strategy name. We use $\geq_s^K$ to denote the preorder on models defined by (s, K). We first introduce the following notation and auxiliary definitions:

$$
\begin{array}{ll}
K^n(m) & = \{f \mid (f, n) \in K, m \models f\} \\
maxsat^K(m) & = -\infty \text{ if } m \not\models f_i \text{ for all } (f_i, v_i) \in K, \\
 & \quad max\{i \mid (f, i) \in K, m \models f\} \text{ otherwise.} \\
maxunsat^K(m) & = -\infty \text{ if } m \models f_i \text{ for all } (f_i, v_i) \in K, \\
 & \quad max\{i \mid (f, i) \in K, m \not\models f\} \text{ otherwise.}
\end{array}
$$

Now we can define the corresponding orderings on models:

- $m_1 \geq_\top^K m_2$ iff $maxsat^K(m_1) \geq maxsat^K(m_2)$.
- $m_1 \geq_\kappa^K m_2$ iff $maxunsat^K(m_1) \leq maxunsat^K(m_2)$.
- $m_1 \geq_\subseteq^K m_2$ iff $K^n(m_1) = K^n(m_2)$ for all n, or there is an n such that $K^n(m_1) \supset K^n(m_2)$, and for all $j > n$: $K^j(m_1) = K^j(m_2)$
- $m_1 \geq_\#^K m_2$ iff $|K^n(m_1)| = |K^n(m_2)|$ for all n, or there is an n such that $|K^n(m_1)| > |K^n(m_2)|$, and for all $j > n$: $|K^j(m_1)| = |K^j(m_2)|$

The strategies can be described informally as follows:

- $\top$ prefers m_1 over m_2 whenever the most important goal satisfied by m_1 is more important than the most important goal satisfied by m_2. It was used in (Benferhat *et al.* 2002) in the context of bipolar representations. With this strategy the intuitive reading of (f, r) is: if f is true, then the total satisfaction is at least r.
- κ prefers m_1 over m_2 whenever the most important goal not satisfied by m_1 is less important than the most important goal not satisfied by m_2, in other words, if the rank r such that all goals of rank r and higher are satisfied is lower in m_1 than the corresonding rank in m_2. This is the κ-ranking used in system Z. It is also the ordering needed to model a rejection p (Benferhat *et al.* 2002) via the goal $\neg p$.
- to check whether $\subseteq$ prefers m_1 over m_2 we start from the most important goals and go down stepwise to less important ones. If, at the first rank reached this way for which the formulas satisfied by the two models differ, we have that m_1 satisfies a superset of the formulas satisfied by m_2, then m_1 is preferred. This is the order used in (Brewka 1989).
- $\#$ is similar to $\subseteq$, but rather than checking the sets of formulas satisfied for each rank, their cardinality is considered. This is the proposal of Benferhat and colleagues in (Benferhat *et al.* 1993).

Among the preorders on models generated by these strategies only $\geq_\subseteq^K$ is partial. The others are total, that is, the ordering on models is again a ranking.

To illustrate the strategies let us consider the following RKB:

$$K = \{(a, 2), (b, 2), (c, 2), (d, 1), (e, 1)\}$$

We will represent models by a sequence of atoms true in the model. For example, acd represents the model in which a, c and d are true, b and e are false. Also, whenever K is clear from context we omit the upper index K from the relation symbols. We have $ad >_\top de$ since ad, contrary to de, satisfies a goal of rank 2. On the other hand, $ad \not>_\kappa de$ since both models falsify a goal of rank 2. Furthermore, $abc >_\kappa bd$ since abc satisfies all goals of rank 2, that is, the maximal rank of a violated goal is 1. On the other hand $abc \not>_\top bd$ since both satisfy a goal of rank 2. abd is incomparable to cd according to $\subseteq$, however $abd >_\# cd$ since the former satisfies two goals of rank 2.

The different strategies are not independent of each other. We have the following results:

Proposition 2 *Let m_1 and m_2 be models, K a ranked knowledge base. The following relationships hold:*

$$
\begin{array}{l}
m_1 >_\top^K m_2 \textit{ implies } m_1 >_\subseteq^K m_2, \\
m_1 >_\kappa^K m_2 \textit{ implies } m_1 >_\subseteq^K m_2, \\
m_1 >_\subseteq^K m_2 \textit{ implies } m_1 >_\#^K m_2, \\
m_1 \geq_\subseteq^K m_2 \textit{ implies } m_1 \geq_\#^K m_2, \\
m_1 >_\top^K m_2 \textit{ implies } m_1 \geq_\kappa^K m_2, \\
m_1 >_\kappa^K m_2 \textit{ implies } m_1 \geq_\top^K m_2.
\end{array}
$$

The first 4 relationships can be illustrated using the following figure:

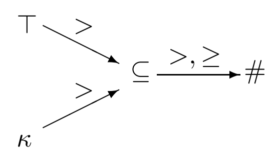

Fig.1: Relationship among basic orderings

More relationships can be established if we allow K to be modified.

Proposition 3 *Let K be a ranked knowledge base, m_1 and m_2 models. Let $K_\wedge =$*

$$\{(C_i, i) \mid C_i \textit{ conjunction of all } f \textit{ with } (f, j) \in K, j \geq i\}$$

and $K_\vee =$

$$\{(C_i, i) \mid C_i \textit{ disjunction of all } f \textit{ with } (f, j) \in K, j \geq i\}.$$

Then $m_1 \geq_\kappa^K m_2$ iff $m_1 \geq_\subseteq^{K_\wedge} m_2$ and $m_1 \geq_\top^K m_2$ iff $m_1 \geq_\subseteq^{K_\vee} m_2$.

Moreover, since $\subseteq$ and $\#$ are equivalent if for each rank there is only a single formula possessing this rank, the proposition also holds if we use $\#$ instead of $\subseteq$.

The preference language

So far we discussed basic preference descriptions only. A user may have different ways of modeling her preferences for different aspects of a problem. Therefore, we also want to allow more complex descriptions representing combinations of the corresponding preorders.

We now give the full definition of our logical preference description language. For reasons which will become clear later we use the standard propositional connectives together with a new connective $>$ expressing preference among expressions.

Definition 4 *The logical preference description language LPD is inductively defined as follows:*

1. *each basic preference description is in LPD,*
2. *if d_1 and d_2 are in LPD, then the expressions $(d_1 \wedge d_2)$, $(d_1 \vee d_2)$, $(d_1 > d_2)$ and $-d_1$ are in LPD.*

The formal definition of the meaning of a (non-basic) LPD expression, that is the definition of its associated preorder on models, is as follows:

Definition 5 *Let R_1 and R_2 be the preorders on models represented by d_1 and d_2, respectively. Let $tr(R)$ denote the transitive closure of a relation R. $Ord(lpd)$, the preorder represented by the complex LPD expression lpd, is defined as follows:*

$$
\begin{aligned}
Ord(d_1 \wedge d_2) &= R_1 \cap R_2 \\
Ord(d_1 \vee d_2) &= tr(R_1 \cup R_2) \\
Ord(-d_1) &= \{(m_2, m_1) \mid (m_1, m_2) \in R_1\} \\
Ord(d_1 > d_2) &= \{(m_1, m_2) \in R_1 \mid (m_1, m_2) \in R_2 \textit{ or } \\
&\qquad\qquad (m_2, m_1) \notin R_1\}
\end{aligned}
$$

$d_1 \wedge d_2$ corresponds to the well-known Pareto ordering: a model m_1 is at least as good as m_2 if it is at least as good as m_2 with respect to both d_1 and d_2. m_1 is strictly better if it is better according to one of the suborderings, and at least as good as m_2 with respect to the other. The definition for $d_1 \vee d_2$ needs the transitive closure since the union of two orderings is not necessarily transitive. The $-$ operator just reverses the original ordering. Double application of $-$ obviously gives back the original ordering. Note, however, that other properties of negation do not hold for $-$, in particular the de Morgan laws do not hold. For instance, $-(d_1 \vee d_2)$ differs from $(-d_1 \wedge -d_2)$.[4]

$d_1 > d_2$ is the lexicographic ordering of R_1 and R_2 which gives more priority to R_1 and uses R_2 only to distinguish between models which are equally good wrt. R_1. Here, m_1 is strictly better than m_2 if it is strictly better wrt. R_1, or as good as m_2 wrt. R_1 and strictly better wrt. R_2.

The binary operators $\vee, \wedge$ and $>$ are associative. We omit brackets if this does not cause confusion, assuming binding strength decreases in the order $\wedge, \vee, >$.

The language LPD gives us flexible means of representing preferences on models. We next discuss some properties of the language.

Under certain circumstances expressions can be simplified. We say a preference expression d_1 implies an expression d_2 iff $Ord(d_1) \subseteq Ord(d_2)$. We say two preference expressions are equivalent iff they induce the same preorder on models, that is, iff $Ord(d_1) = Ord(d_2)$. For instance, let $s \in Strat$ be any of our basic strategies, then the expression:

$$(\{(f_1, r_1), \ldots, (f_n, r_n)\}^s > \{(s_1, r_1'), \ldots (s_m, r_m')\}^s)$$

is equivalent to

$$\{(f_1, c + r_1), \ldots, (f_n, c + r_n), (s_1, r_1'), \ldots (s_m, r_m')\}^s$$

where $c = max\{r_i'\} + 1$.

[4] $-(d_1 \vee d_2)$ is equivalent to $(-d_1 \vee -d_2)$, and $-(d_1 \wedge d_2)$ equivalent to $(-d_1 \wedge -d_2)$, though.

Note that this result depends on the fact that the two basic preference expressions use the same strategy. A similar result for different strategies does not hold. Also, for $\wedge$ such simplifications are not possible, even if the strategies of the subexpressions coincide. The only weak result we get is:

Proposition 6 *Let K_1 and K_2 be RKBs. $(K_1^{\subseteq} \wedge K_2^{\subseteq})$ implies $(K_1 \cup K_2)^{\subseteq}$.*

The other direction does not hold (to see this, consider the case where we split an RKB such that formulas with high rank are in K_1, formulas with low rank in K_2). For the cardinality based strategy, using the union of 2 RKBs, that is $(K_1 \cup K_2)^{\#}$, clearly is different from $(K_1^{\#} \wedge K_2^{\#})$. In the general case complex expressions are not reducible to single ones which use the same formulas, even if the ranks are allowed to change.

Example: Selecting a Movie

In this section we want to illustrate the use of our language with a commonsense example. Assume you are planning to go to the cinema with your girl friend. Both of you prefer comedies over action movies over tragedies. Your girl friend loves to see Hugh Grant and Brad Pitt, followed by Leonardo di Caprio. Your favourite actors are Julia Roberts and Nicole Kidman, followed by Gwyneth Paltrow and Halle Berry. You both feel that the type of movie is as important as the actors. Moreover, since it is your girl friend's birthday, her actors' preferences are more important today than yours.

We can represent this information using the following RKBs:

$$
\begin{aligned}
&K_1 = \{(Hugh, 2), (Brad, 2), (Leo, 1)\} \\
&K_2 = \{(Julia, 2), (Nicole, 2), (Gwyneth, 1), (Halle, 1)\} \\
&K_3 = \{(comedy, 3), (action, 2), (tragedy, 1)\}
\end{aligned}
$$

We assume the background knowledge contains information that the mentioned types of movies are mutually exclusive, models thus will make at most one of the types true.

Since seeing more of the favourite actors is more fun we use the cardinality based strategy. Our preferences can thus be represented as the LPD expression:

$$(K_1^{\#} > K_2^{\#}) \wedge K_3^{\top}$$

Now assume we have the following information about the movies shown tonight:

$$
\begin{aligned}
&M_1 : comedy, Hugh, Brad \\
&M_2 : comedy, Hugh, Leo, Julia \\
&M_3 : comedy, Brad, Leo, Julia, Halle \\
&M_4 : action, Brad, Hugh, Nicole \\
&M_5 : action, Brad, Leo, Julia, Halle \\
&M_6 : tragedy, Brad, Leo, Julia, Nicole
\end{aligned}
$$

We assume that the list of actors mentioned for each movie is complete, that is, if one of the names appearing in the RKBs is not listed, then this actor is not in the corresponding movie.

We represent the information listed above in the background knowledge in the form of logical implications. For instance, for M_1 we get:

$$M_1 \rightarrow\ comedy \\ \wedge\ Hugh \wedge Brad \wedge \neg Leo \\ \wedge \neg Julia \wedge \neg Nicole \wedge \neg Gwyneth \wedge \neg Halle$$

We also represent that exactly one of the 6 movies needs to be chosen, that is exactly one of $\{M_1, \dots, M_6\}$ must be true in each model. All models thus contain one selected movie together with its type and its actors.

According to our preference expression, M_1 is preferred over M_2 and over M_3 because two of your girl friend's most favourite actors play in M_1. M_3 is preferred over M_2 since it is as good with respect to your girl friend's preferences (trading Hugh for Brad), but better according to your preferences since it additionally gives you Halle.

M_4 and M_1 are incomparable: M_1 is the better type of movie, but M_4 is better with respect to its actors. M_5 is worse than both M_4 (worse actors according to your girl friend) and M_3 (worse type), and thus also worse than M_1. M_6 is less preferred than both M_4 and M_1: it has less preferred actors and a worse type. M_6 is incomparable to M_5.

The only non-dominated movies are thus M_1 and M_4. The preference structure among models (represented through the selected movies) is illustrated in the following figure (arrows point to strictly preferred models):

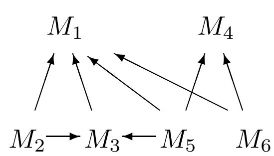

Fig.2: Strict preferences among movies

Discussion

In this paper we developed a flexible preference representation language. The basic building blocks of the language are ranked knowledge bases together with a model selection strategy. Ranked knowledge bases allow us to represent prioritized goals conveniently. We investigated four different strategies known from the literature, all of them qualitative in the sense that the induced total preorder on formulas is what counts, rather than the actual numbers.

Our language also allows for combinations of preference expressions. Conjunction naturally leads to Pareto orderings based on the underlying subexpressions. The connective $>$ allows us to define lexicographic orderings. The language also has disjunction and a form of negation which simply reverses the original order.

The work presented in this paper shares some motivation with (Brewka 2004). Also in that paper a language, called PDL, for expressing complex preferences is presented. However, there are several major differences which are due to the fact that PDL is taylored towards answer set optimization:

1. PDL is rule based rather than goal based. The basic building blocks are rules with prioritized heads rather than ranked knowledge bases.
2. Since PDL is used to assess the quality of answer sets (i.e., sets of literals) rather than models, it becomes important to distinguish between an atom not being in an answer set and its negation being in an answer set. In other words, the distinction between classical negation and default negation (negation as failure) is relevant. Since we are interested in preferences among models here, this distinction does not play a role in LPD.
3. PDL distinguishes between penalty producing and other strategies. Both numerical and qualitative combination strategies are thus used. On the other hand, combinations corresponding to our disjunction and negation operators are lacking.

Although we restricted our discussion to purely qualitative approaches, there is no principle obstacle against integrating numerical approaches as well, at least at the level of basic preference expressions. For instance, we could use ranks as penalties or rewards and define the preorder on models on the basis of the actual rank values. The reader should be aware, though, that this only works on the basic level. The connectives we defined operate on the preorders and do not take numerical information into account. Any numerical information would thus be lost in our language at the level of complex preference expressions.

An interesting related paper is (Son & Pontelli 2004) which introduces a preference language for planning. The language is based on a temporal logic and is able to express preferences among trajectories. As in LPD, preferences can be combined via binary operators - somewhat different from ours. The major difference certainly is that our approach aims at being application-independent, whereas (Son & Pontelli 2004) is geared towards planning.

Also related is (Andreka, Ryan, & Schobbens 2002). The authors investigate combinations of priority orderings based on a generalized lexicographic combination method. This method is more general than usual lexicographic orderings - including the ones expressible through our $>$ operator - since it does not require the combined orderings to be linearly ordered. It is based on so-called priority graphs where the suborderings to be combined are allowed to appear more than once. The authors also show that all orderings satisfying certain properties derived from Arrow's conditions (Arrow 1950) can be obtained through their method. This is an interesting result. On the other hand, we found it somewhat difficult to express examples like our movie example using the method. We believe our language is closer to the way people actually describe their preferences.

In (Boutilier *et al.* 1999) CP-networks are introduced, together with corresponding algorithms. These networks are a graphic representation, somewhat reminiscent of Bayes nets, for conditional preferences among feature values under the *ceteris paribus* principle. Our approach differs from CP-networks in several respects: (1) Preferences in CP-networks are always total orders of the possible values of a single variable. We are able to represent arbitrary prioritized goals. (2) The ceteris paribus interpretation of preferences is very different from our goal-based interpretation. The former views the available preferences as (hard) constraints on a global preference order. Each preference relates only models which differ in the value of a single variable. A set of ranked goals, on the other hand, is more like a set of different criteria in multi-criteria optimization. In particular, goals can be conflicting. Conflicting goals may neutralize each other, but do not lead to inconsistency.

Although our work was mainly motivated by several approaches developed in the area of nonmonotonic reasoning, many related ideas can be found in constraint satisfaction, in particular valued (sometimes also called weighted) constraint satisfaction (Freuder & Wallace 1992; Fargier, Lang, & Schiex 1993; Schiex, Fargier, & Verfaillie 1995; Bistarelli, Montanari, & Rossi 1997). A valued constraint, rather than specifying hard conditions a solution has to satisfy, yields a ranking of solutions. A global ranking of solutions then is obtained from the rankings provided by the single constraints through some combination rule. This is exactly what happens in our approach on the level of basic preference expressions. Also in constraint satisfaction we find numerical as well as qualitative approaches. In MAX-CSP (Freuder & Wallace 1992), for instance, constraints assign penalties to solutions, and solutions with the lowest penalty sum are preferred. In fuzzy CSP (Fargier, Lang, & Schiex 1993) each solution is characterized by the worst violation of any constraint. Preferred solutions are those where the worst violation is minimal. This corresponds to the κ strategy. We are not aware of any approach in constraint satisfaction trying to combine different strategies. For this reason we believe the language developed here will be of interest also for the constraint community.

In future work we plan to investigate the use of partially ordered rather than ranked knowledge bases on the level of basic preference expressions. We also plan to investigate computational issues related the approach. In particular, it would be interesting to see whether a generate and improve method like the one developed for answer set optimization in (Brewka 2004) can be used here as well.

Acknowledgements

The paper greatly benefitted from discussions with and suggestions by Jérôme Lang. The author acknowledges support from DFG (Computationale Dialektik: BR 1817/1-5).

References

Andreka, H.; Ryan, M.; and Schobbens, P.-Y. 2002. Operators and laws for combining preference relations. *Journal of Logic and Computation* 12(1):13–53.

Arrow, K. 1950. A difficulty in the concept of social welfare. *Journal of Political Economy* 58:328–346.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In *Proceedings International Joint Conference on Artificial Intelligence, IJCAI-93*, 640–645. Morgan Kaufmann.

Benferhat, S.; Dubois, D.; Kaci, S.; and Prade, H. 2002. Bipolar representation and fusion of preferences in the possibilistic logic framework. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*, 421–448. Morgan Kaufmann.

Benferhat, S.; Dubois, D.; and Prade, H. 2002. Representing default rules in possibilistic logic. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning, KR-92, Cambridge, n2002*, 673–684. Morgan Kaufmann.

Bistarelli, S.; Montanari, U.; and Rossi, F. 1997. Semiring-based constraint solving and optimization. *Journal of the ACM* 44(2):201–236.

Boutilier, C.; Brafman, R.; Hoos, H.; and Poole, D. 1999. Reasoning with conditional ceteris paribus preference statements. In *Proc. Uncertainty in Artificial Intelligence, UAI-99*.

Brewka, G. 1989. Preferred subtheories - an extended logical framework for default reasoning. In *Proc. International Joint Conference on Artificial Intelligence, IJCAI 89*, 1043–1048.

Brewka, G. 2004. Complex preferences for answer set optimization. In *Proceedings 9th International Conference on Principles of Knowledge Representation and Reasoning, KR-04*. Morgan Kaufmann.

Fargier, H.; Lang, J.; and Schiex, T. 1993. Selecting preferred solutions in fuzzy constraint satisfaction problems. In *Proceedings of the First European Congress on Fuzzy and Intelligent Technologies*.

Freuder, E., and Wallace, R. 1992. Partial constraint satisfaction. *Artificial Intelligence* 58(1):21–70.

Goldszmidt, M., and Pearl, J. 1991. System Z+: A formalism for reasoning with variable-strength defaults. In *Proc. 9th National Conference on TArtificial Intelligence*, 399–404. Morgan Kaufmann.

Lang, J. 2004. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence* to appear.

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to default reasoning. In Vardi, M., ed., *Proc. 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, 121–135. Morgan Kaufmann.

Schiex, T.; Fargier, H.; and Verfaillie, G. 1995. Valued constraint satisfaction problems: Hard and easy problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI-95*, 631–637.

Son, R., and Pontelli, E. 2004. Planning with preferences using logic programming. In *Proc. 7th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 04*, 247–260. Springer LNAI 2923.