IPMU 2004, July 4-9, 2004
Perugia, Italy

Management of uncertainty orderings through ASP*

Andrea Capotorti
Dipartimento di Matematica e Informatica
Università di Perugia, Italy
capot@dipmat.unipg.it

Andrea Formisano
Dipartimento di Informatica
Università dell'Aquila, Italy
formisano@di.univaq.it

Abstract

In this paper we exploit Answer Set Programming (ASP) to formalize (and reason about) uncertainty expressed by belief orderings. The availability of ASP-solvers supports the design of automated tools to handle such formalizations. Our proposal reveals particularly suitable whenever the domain of discernment is "partial", i.e. it does not represent a closed world but just the relevant part of a problem.
We first illustrate how to automatically "classify", according to the most well-known uncertainty frameworks, any given partial qualitative uncertainty assessment. Then, we show how to compute the enlargement of an assessment to any other new inference target, with respect to a fixed (admissible) qualitative framework.

Key words: Uncertainty orderings, answer set programming, partial assessments, general inference.

Introduction

Uncertainty orderings are receiving wider and wider attention in AI literature, either as theoretical tools to deal directly with belief management [1, 6, 7], or inside the more articulated framework of decision-making theory (see, for example, [8, 9, 10, 11, 14]). This burst of interest translates into a wider application of uncertainty orderings. This is because qualitative assessments better fit the nature of human judgments, while they do not suffer intrinsic difficulties typical of numerical elicitations.

Similarly to what happened for numerical methods, there have been proposed various uncertainty orderings different from traditional probabilistic ones. The main purpose of all of these proposals is to generalize the "additive" character of (unconditional) probabilities. As usual, different generalizations are possible, hence several families of qualitative judgments appeared in literature. As a matter of fact, all of them share common basic properties, while they differentiate on the way of combining pieces of information.

By following the way paved by [5, 7, 18, 19], in [2, 3, 4] various preference orderings have been fully classified according to their agreement with the most well-known numerical models; both for "complete" (i.e. defined on well structured and closed supports) and "partial" assessments (i.e. defined only among some of the quantities involved).

Such axiomatic classifications revealed a coarser grained differentiation with respect to numerical models: different uncertainty measures share the same qualitative properties in combining pieces of information.

*Research partially funded by MIUR ex-40% project *Aggregate- and number-reasoning for computing: from decision algorithms to constraint programming with multisets, sets, and maps* and by the *Information Society Technologies programme of the European Commission, Future and Emerging Technologies* under the IST-2001-37004 WASP project.

The detected classes have been characterized by identifying specific axioms that the orderings must satisfy. Apart from qualitative probabilities, such axioms are of direct declarative reading, as they involve only logical and preference relations (among events). As we will see, such a declarative character supports a straightforward translation of the axioms within the logical framework of Answer Set Programming (ASP). As a consequence, we immediately obtain an executable ASP-specification able to discriminate between the different uncertainty orderings. More specifically, this is done by exploiting an ASP-solver (in our case Smodels, cf. [20]) that determines the set of axioms that are violated by the given preference relation, which expresses the user's believes comparisons.

Notice that, by proceeding in this way, we actually invert the usual attitude towards qualitative management of uncertainty. In fact, specific axioms are usually set in advance, so that only relations satisfying them are admitted. Here, on the contrary, given a fixed preference relation, our goal consists in ascertain which are the reasonable rules to work with.

In this paper, we mainly focus on the treatment of partial orderings, even if also total relations can be dealt with by using the very same machinery. The rational behind this choice is that incomplete assessments appear more realistic models of phenomena which are open to include new (and maybe unexpected) evidences or considerations.

In this frame of mind, an interesting problem is that of finding an extension of a preference relation so as to take into account any further event extraneous "in some sense" to the initial assessment. Obviously, this should be achieved in a way that the initial ordering and its extension satisfy the same axioms. In other words, first, the given (partial) preference relation is classified by identifying its characterizing rules (as usual, in case of multiple possibilities the most restrictive axioms are singled out). Then, the very same approach based on declarative programming is adopted in order to determine the extension sought for.

The paper is organized as follows: next section briefly describes the axioms characterizing partial ordinal uncertainty relations. Sec. 2 recalls the main features of ASP, with particular emphasis on the application of ASP to the above mentioned issues. In Sections 3 and 4 we illustrate, also by simple examples, the potentialities of the our approach. Finally, we draw conclusions and outline future developments.

1 Axioms characterizing partial uncertainty orderings

In [2, 3, 4] axioms have been introduced to distinguish alternative possible approaches to qualitatively manage uncertainty. In particular, according to the major uncertainty frameworks, a full classification of partial orderings has been proposed in [4]. In the work we are going to present, we face a problem left open in [4]. Namely, we propose an inferential system to classify a given partial ordinal relation into one of the classes.

Let us start by briefly recalling the basic notions on uncertainty orderings and on their axiomatic classification. We will not enter into the details of the motivations for such classification, the reader is referred to [2, 3, 4].

The domain of discernment is represented by a finite set of events $\mathcal{E} = \{E_1, \ldots, E_n\}$ (among them, ϕ and Ω denote the impossible and the sure event, respectively). The events in $\mathcal{E}$ are seen as the relevant propositions on which the subject of the analysis can (or wants) to express his opinion. Hence usually $\mathcal{E}$ does not represent a full model, i.e. it does not comprehend all elementary situations and all of their combinations. For this, a crucial component of partial assessments is the knowledge of the logical relationships (incompatibilities, implications, combinations, equivalences, etc.) holding among the events E_is. Such constraints are usually represented as a set $\mathcal{C}$ of clauses among the E_is.

Taking into account the constraints $\mathcal{C}$, the family $\mathcal{E}$ spans a minimal Boolean algebra $\mathcal{A}_\mathcal{E}$ containing $\mathcal{E}$ itself. Note that $\mathcal{A}_\mathcal{E}$ is only implicitly defined via $\mathcal{E}$ and $\mathcal{C}$ and it is not a part of the assessment. Anyway, $\mathcal{A}_\mathcal{E}$ will be referenced as a supporting tool.

Let $\preceq$ be a partial (i.e. not necessarily defined for all pairs (A, B) in $\mathcal{E} \times \mathcal{E}$) ordinal relation among events expressing the intuitive idea of being "less or equal than" or "not preferred to". The symbols $\sim$ and $\prec$ denote the symmetrical part and asymmetrical part of $\preceq$, respectively.

The mainline property usually required to $\preceq$ is to be compatible with a capacity (a function monotone with respect to $\subseteq$, thought to be a "basic" property for any uncertainty measure). More precisely, $\preceq$ should be extensible to a full relation $\preceq^*$ over $\mathcal{A}_\mathcal{E} \times \mathcal{A}_\mathcal{E}$ representable by a capacity.

Recall that, in general, a function f from $\mathcal{A}_\mathcal{E}$ to $\mathbb{R}$ represents (or, equivalently, induces) a full relation $\preceq^*$ if, for each pair (A, B) we have

$$A \preceq^* B \Longleftrightarrow f(A) \leq f(B).$$

The basic compatibility requirement translates into the following axioms, shared by all reasonable partial orderings: $\preceq$ must be a reflexive binary relation on $\mathcal{E}$ such that

(A1$'$) there are no intransitive cycles;

(A2$'$) $\neg(\Omega \preceq \phi)$;

(A3$'$) for all $A, B \in \mathcal{E}$, $A \subseteq B \Longrightarrow \neg(B \prec A)$

where $\neg(B \prec A)$ means that the pair (B, A) does not belongs to $\preceq$.

Mathematical properties of ordinal relations satisfying basic axioms (A1$'$), (A2$'$) and (A3$'$) are deeply investigated in [6].

In the sequel, we consider these axioms as prerequisites for any investigation on $\preceq$. Differentiation among ordinal relations can be done on the basis of more specific way of combining different pieces of information. Below, we list the axioms characterizing each class.[1] The name of the classes comes from the representability of $\preceq^*$ at least by corresponding numerical measures (observe that qualitative classes are coarser than numerical measures).[2]

Comparative probabilities. The ordering $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by a probability function or by a λ-measure* iff

(CP) for any $A_1, \ldots, A_n, B_1, \ldots, B_n \in \mathcal{E}$, with $B_i \preceq A_i, \forall i = 1, \ldots, n$, such that for some $r_1, \ldots, r_n > 0$

$$\sup \sum_{i=1}^n r_i(a_i - b_i) \leq 0$$

implies that $A_i \sim B_i$, for all $i = 1, \ldots, n$ (a_i, b_i denote the indicator functions of A_i, B_i, resp.).

Comparative believes. $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by a belief function or by a n-monotone function* (with $n \geq 2$) iff for all $A, B, C \in \mathcal{E}$ s.t. $A \subset B$, $B \wedge C = \phi$ it holds

(B$'$) $A \prec B \Longrightarrow \neg(B \vee C \preceq A \vee C)$.

Comparative lower-probabilities. $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by a lower-probability function or by a 0-monotone function*[3] iff for all $A, B \in \mathcal{E}$ s.t. $A \wedge B = \phi$ it holds

(L$'$) $\phi \prec A \Longrightarrow \neg(A \vee B \preceq B)$.

Comparative plausibilities. $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by a plausibility function or by a n-alternating function* (with $n \geq 2$) iff for all $A, B, C \in \mathcal{E}$ s.t. $A \subset B$ it holds

(PL$'$) $A \sim B \Longrightarrow \neg(A \vee C \prec B \vee C)$.

Comparative upper-probabilities. $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by an upper-probability function or by a 0-alternating function*[4] iff for all $A, B, C \in \mathcal{E}$ s.t. $A \wedge B = \phi$ it holds

(U$'$) $\phi \sim A \Longrightarrow \neg(C \prec A \vee C)$.

Comparative lower/upper-probabilities. $\preceq$ can be simultaneously extended to a full ordinal relation $\preceq^*$ *representable by both a*

---

[1] Note that we characterize each class by a single axiom, whereas in [4] some classes are described by introducing further axioms. It is easy to see that these further axioms are redundant whenever we consider to enlarge $\prec$ by monotonicity (i.e. $A \subseteq B \Longleftrightarrow A \prec B$).

[2] Axiom (CP) was originally introduced in [5]. Axiom (B$'$) derives by the analogous axiom introduced for complete orderings in [19].

[3] "0-monotone" functions are also known as "super-additive" i.e. $f(H \vee K) \geq f(H) + f(K)$ for all $H, K \in \mathcal{A}_\mathcal{E}$ s.t. $H \wedge K = \phi$.

[4] "0-alternating" functions are also known as "sub-additive" i.e. $f(H \vee K) \leq f(H) + f(K)$ for all $H, K \in \mathcal{A}_\mathcal{E}$ s.t. $H \wedge K = \phi$.

lower-probability function and *by an upper-probability function* iff it contemporarily satisfies axioms (L′) and (U′).

Note that only axiom (CP) does not have a pure qualitative nature since it involves also indicator functions and summations. It is the only one that should require an external numerical elaboration (e.g. involving some linear programming tool such as the simplex or the interior point methods). Meanwhile, to remain within the same type of axioms, the following *necessary* axiom (WC) can also be considered. Note that (WC), if taken by itself, does not guarantee the representability of $\preceq^*$ by a probability function; nevertheless, its falure witnesses non-representability.

***weak* comparative probabilities.** If $\preceq$ can be extended to a full ordinal relation $\preceq^*$ *representable by a probability function* then for all $A, B, C \in \mathcal{E}$ s.t. $A \wedge C = B \wedge C = \phi$
(WC) $\quad A \preceq B \Longrightarrow \neg(B \vee C \prec A \vee C)$

All qualitative axioms are of direct reading, i.e. they explicit which are the rules to follow in combining elements of the domain $\mathcal{E}$ to remain inside a specific framework.

2 Answer set programming

Answer Set Programming (ASP) is an emergent, alternative style of logic programming [17, 15] based on the Answer Set (or equivalently Stable Model) semantics [12, 13]: each solution to a problem is represented by an *answer set* of a function-free logic program encoding the problem itself. The Answer Set semantics is a view of logic programs as sets of (default) inference rules. Alternatively, one can see a program as a set of constraints on the solution of a problem, where each answer set represents a solution compatible with the constraints constituting the program. Consider for instance the simple logic program $\{q \leftarrow \textit{not } p. \quad p \leftarrow \textit{not } q.\}$, where the first of the two rules is read as "assuming that p is false, we can *conclude* that q is true." This program has two answer sets. In the first one, q is true whereas p is false; in the second one, p is true whereas q is false. Then, in ASP we are able to manage cyclic negative dependencies that represent incomplete knowledge or denote the possibility of different alternatives, by representing each consistent choice by means of an answer set. Notice that an ASP-program may have none, one, or several answer sets.

To solve a problem using ASP means to write a set of rules whose answer sets correspond to solutions, and then find a solution using an ASP-solver [20]. The basic approach to writing such a program is known as the "generate-and-test" strategy. First one writes a group of rules for defining "potential solutions" i.e. an easy-to-describe superset of the set of solutions. Then one adds a group of constraints that rule out the potential solutions that are not solutions.

Consider, for instance, the use of this method for finding the three-colorings of a graph (the program below is in the syntax of the Smodels solver).[5] The statement `node(0..3)` is a shortcut for the definition of a set of facts, namely `node(0),...,node(3)`. The symbol "|" denotes disjunction, and allows us to state that a node can be assigned one of the three colors `red`, `blue`, `green`, that are introduced by facts `col(·)`. This defines all possible colorings of the graph.

```
col(red).     col(blue).    col(green).
node(0..3).   edge(0,1).    edge(1,2).
edge(2,0).    edge(2,3).    edge(1,3).
color(X,red) | color(X,blue) | color(X,green) :- node(X).
```

It remains to be stated that we wish to select only the colorings where adjacent nodes have a different colors. This is done by introducing a *constraint* (i.e. a rule with empty consequent), whose conditions cannot be all true, otherwise they would imply falsity. In particular, the first of the following rules states that there cannot be two adjacent nodes (nodes connected by an edge) to which the same color is assigned:

```
:- edge(X,Y), col(C), color(X,C), color(Y,C).
1{color(X,C): col(C)}1 :- node(X).
```

On the other hand, the rule with head in brackets, called *weight constraint*, imposes that for all `X` which is a node, the property

[5] In the syntax of Smodels ':-' denotes implication ←, while ',' stands for conjunction.

color(X,C) can take one and only one value for C, among those defined by predicate col(C) (see [16] for a description of the general form of this kind of constraint).

When fed with this program, Smodels (and, similarly, any of the solvers) computes six answer sets. Each of them corresponds to one legal 3-coloring of the graph. The following is one of such answer sets:

```
Answer: 1. Stable Model:
color(0,green) color(1,blue) color(2,red) color(3,green).
```

3 Preference classification

Our first task consists in writing an ASP-program able to classify any given partial ordering $\preceq$, w. r. t. the axioms seen in Sec. 1 (except for (CP), that, up to our knowledge, does not admit a purely declarative formulation). A preliminary step is the introduction of suitable predicates, namely, prec(·,·), precneq(·,·), and equiv(·,·), to render in ASP the relators $\preceq$, $\prec$, and $\sim$, respectively. Moreover, the fact of "being an event" (i.e. a member of $\mathcal{E}$) is stated through the monadic predicate event(·).[6] Auxiliary predicates/functions are defined to render usual set-theoretical constructors, such as $\cap$, $\cup$, and $\subseteq$.

The characterization of potential legal answer sets is done by asserting properties of prec(·,·), precneq(·,·), and equiv(·,·), by means of the following rules:

```
prec(E1,E2) :- event(E1), event(E2), equiv(E1,E2).
prec(E2,E1) :- event(E1), event(E2), equiv(E1,E2).
equiv(E1,E2) :- event(E1), event(E2), prec(E2,E1),
                prec(E1,E2).
prec(E1,E2) :- event(E1), event(E2), precneq(E1,E2).
:- precneq(E1,E2), event(E1), event(E2), equiv(E1,E2).
```

Also axioms (A1′), (A2′), and (A3′) must be imposed. For instance (A3′) is rendered by:

```
:- event(E1), event(E2), subset(E1,E2), precneq(E2,E1).
```

which rules-out all answer sets in which there exist two events E_1 and E_2 such that both $E_1 \subseteq E_2$ and $E_2 \prec E_1$ hold.

Consider now one of the axioms of Sec. 1, say (B′), for simplicity. Since we do not want to impose such axiom, but we just want to test whether it is satisfied by the preference relation at hand, we introduce a rule of the form:

```
failsB1 :- event(A), event(B), event(C), subset(A,B),
           A!=B, empty(interset(B,C)), precneq(A,B),
           prec(unionset(B,C),unionset(A,C)).
```

whose meaning is that the fact failsB1 will be true (i.e. will belong to the answer set) whenever there exist events falsifying axiom (B′). Analogous treatment has been done for the other axioms (L′), (U′), (PL′), and (WC).

When Smodels is fed with such program, together with a description of an input preference relation, two results may be produced:

a) no answer set exists. This means that the particular preference relation violates some basic requirement, such as axioms (A1′), (A2′), or (A3′).

b) An answer set exists. In this case, the presence of a fact of the form fails$\mathcal{C}$, say failsL1 for instance, witnesses that the corresponding axiom (L′) is violated. Consequently, no total extensions of the given ordering can be represented by a comparative lower-probability.

Example 1 Suppose a physician wants to perform a preliminary evaluation about the reliability of a test for SARS (Severe Acute Respiratory Syndrome). Up to his knowledge, the SARS diagnosis is based on moderate or severe respiratory symptoms and on the positivity or indeterminacy of an adopted clinical test about the presence of the SARS-associated antibody coronavirus (SARS-CoV). Elements appearing in his analysis can be schematized by:

A $\equiv$ *Normal respiratory symptoms*
B $\equiv$ *Moderate respiratory symptoms*
C $\equiv$ *Severe respiratory symptoms*
D $\equiv$ *Moderate or sever respiratory symptoms*
E $\equiv$ *Death from pulmonary diseases*
F $\equiv$ *Positive or indeterminate clinical test*

subject to these (logical) restrictions:

A∩B=∅, B∩C=∅, A∩C=∅, A∪B∪C=Ω, D=A∪B, E⊂C, F∩A=∅.

Consider the following partial ordering:

precneq(∅,C). precneq(C,B). prec(B,A). precneq(C,D). precneq(E,C). precneq(E,D). precneq(F,A). equiv(A∪E,A∪C).

[6] Actually, in our program, events are denoted by integer numbers. Here, for the sake of readability, we systematically denote events by capital letters.

Due to events' meaning, such ordering seems reasonable. In fact, if it is given as input to Smodels, the answer set found includes the facts failsB1 and failsWC. This means that the given preference relation agrees with the basic rules, however it cannot be managed by exploiting neither a probability nor a belief-function (nevertheless, one can use comparative lower-probabilities or comparative plausibilities). ■

4 Ordering extension

Let be given an initial (partial) assessment expressed as a set of known events $\mathcal{E}$ together with a (partial) ordering $\preceq$ over $\mathcal{E}$. Moreover, assume that $\preceq$ satisfies the axioms characterizing a specific class, say $\mathfrak{C}$, of orderings (cf. Sec. 3). Consider now a new event S (not in $\mathcal{E}$), implicitly described by means of a collection $\mathcal{C}'$ of set-theoretical constraints involving the known events. In the spirit of [5, Theorem 3], the problem we are going to tackle is: Determine which is the "minimal" extension $\preceq^+$ (over $\mathcal{E} \cup \{\mathsf{S}\}$) of the given preference relation $\preceq$, induced by the new event, which still belongs to the class $\mathfrak{C}$. In other words, we are interested in ascertaining how the new event S must relate to the members of $\mathcal{E}$ in order that $\preceq^+$ still is in $\mathfrak{C}$.

To this aim we want to determine the sub-collections $\mathcal{L}_\mathsf{S}$, $\mathcal{WL}_\mathsf{S}$, $\mathcal{U}_\mathsf{S}$, and $\mathcal{WU}_\mathsf{S}$, of $\mathcal{E}$ so defined:

$E \in \mathcal{L}_\mathsf{S}$ iff no extension $\preceq^*$ of $\preceq$ can infer that $\mathsf{S} \preceq^* E$

$E \in \mathcal{WL}_\mathsf{S}$ iff no extension $\preceq^*$ of $\preceq$ can infer that $\mathsf{S} \prec^* E$

$E \in \mathcal{U}_\mathsf{S}$ iff no extension $\preceq^*$ of $\preceq$ can infer that $E \preceq^* \mathsf{S}$

$E \in \mathcal{WU}_\mathsf{S}$ iff no extension $\preceq^*$ of $\preceq$ can infer that $E \prec^* \mathsf{S}$

Consequently, any total ordering extending $\preceq$ must, at least, impose that:

$E \prec^+ \mathsf{S}$ for each $E \in \mathcal{L}_\mathsf{S}$,
$E \preceq^+ \mathsf{S}$ for each $E \in \mathcal{WL}_\mathsf{S}$,
$\mathsf{S} \prec^+ E$ for each $E \in \mathcal{U}_\mathsf{S}$, and
$\mathsf{S} \preceq^+ E$ for each $E \in \mathcal{WU}_\mathsf{S}$,

in order to satisfy the axioms characterizing $\mathfrak{C}$.

In what follows, we describe an ASP-program that solves this problem by taking advantage from the computation executed during the classification phase (cf. Sec. 3): it gets as input the knowledge regarding the satisfied axiom, the preference and logical relations on the original set of events. Such program is fed to the solver, together with the description of the new event (see Example 2, below).

The handling of the axioms is done by ASP-rules of the form (here we list the rule for axiom (L$'$), the other axioms are treated similarly):

```
:- holdsL1, event(A), event(B), empty(N),
   empty(interset(A,B)),   precneq(N,A),
   prec(unionset(A,B),B).
```

Such rules (actually, they are constraints, in the sense described in Sec. 2), declare "undesirable" all (total) extensions for which the corresponding axiom is violated. For instance, consider the above rule; whenever the fact holdsL1 is present (i.e. is true in an answer set), then to make the rule satisfied, at least one of the other literals must not belong to the answer set. (Notice that, these literals are all true exactly when (L$'$) is violated.) Consequently, in order to activate this constraint (i.e. to impose axiom (L$'$), for the case at hand) it suffices to add to the input of the solver a fact of the form holdsL1.

A further rule describes which are the potential answer set we are interested in:

```
1{ precneq(E1,E2),
   equiv(E1,E2),
   precneq(E2,E1) }1 :- event(E1), event(E2).
```

This rule simply asserts that any computed answer-set must predicate on each pair E1,E2 of events by stating exactly one, and only one, of the three facts precneq(E1,E2), equiv(E1,E2), and precneq(E2,E1).

Smodels will produce as output the answer sets fulfilling the desired requirements and encoding "legal" total orderings. The collections $\mathcal{L}_\mathsf{S}$, $\mathcal{WL}_\mathsf{S}$, $\mathcal{U}_\mathsf{S}$, and $\mathcal{WU}_\mathsf{S}$ can be obtained by computing the intersection Cn of all these answer sets. (Or, equivalently, by computing the set of logical consequences of the ASP-program. Notice that, in general, Cn needs not to constitute an answer set, by itself.)

Unfortunately, not all the available ASP-solvers offer the direct computation of Cn as a built-in feature (DLV, for instance does, while Smodels does not, cf. [20]). In general, a simple inspection of the answer sets generated by Smodels allows the user to detect which is the minimal extension of the preference relation which is mandatory for each total ordering.

In order to facilitate this detection, we designed a simple post-processor which filters Smodels' output and produces the imposed extension of $\preceq$.

Example 2 Consider the partial ordering introduced in Example 1 and the new event
$\mathsf{S} \equiv$ *The real state of having SARS*
subject to these restrictions:

$$\mathsf{S}{\subset}\mathsf{F} \quad \text{and} \quad \mathsf{F}{\cap}\mathsf{E}{\subset}\mathsf{S}.$$

Since in Example 1 we discovered that the initial preference relation satisfies axiom (PL$'$), we want to impose such axiom and compute the extension of the initial ordering.

Once Smodels' output was filtered, we obtained the following result:[7]

precneq(S,A∪C) precneq(S,A∪E) precneq(S,D)
precneq(S,A) precneq(S,Ω) prec(∅,S) prec(S,F)

showing that, apart from obvious relations induced by monotonicity, no significative constraint involving S can be inferred. Since S and E can be freely compared, this result suggests that either further investigation about relevance of the clinical test or a revision of the initial preference relation, should be performed. ■

Conclusions

With this paper we started to explore the potential benefits of employing ASP for building decision support systems based on qualitative judgments. Thanks to ASP features, the implementation of what could be thought as a kernel of an inference engine, sprouted almost naturally. Next step in this research would consist in validating the proposed approach by means of a number of benchmarks aimed at testing our framework on the ground of real applications. A further goal would consist in completing our approach so that to handle comparative probabilities too. Since no axiomatic characterization of comparative probabilities is known (up to our knowledge), this should be achieved through the exploitation of efficient linear optimization tools (such as the column generation techniques). More in general, we envisage the design of a general tool which integrates different procedures and methods, comprehending mixed numerical/qualitative assessments and conditional frameworks.

[7] We list here only the portion of the extension involving the new event S.

Acknowledgments

We thank Stefania Costantini for fruitful discussions on ASP.

References

[1] T. Bilgiç (2001) Fusing interval preferences. In *Proc. of EUROFUSE Workshop on Preference Modelling and Applications*, Granada, Spain, April 2001, pp. 253–258.

[2] A. Capotorti, G. Coletti, and B. Vantaggi (1998) Non additive ordinal relations representable by lower or upper probabilities. *Kybernetika*, **34**(1):79–90.

[3] A. Capotorti and B. Vantaggi (1998) Relationships among ordinal relations on a finite set of events. In *Information, Uncertainty, Fusion*, Eds. B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh, pp. 447–458.

[4] A. Capotorti and B. Vantaggi (2000) Axiomatic characterization of partial ordinal relations. *Internat. J. Approx. Reason.*, **24**:207–219.

[5] G. Coletti (1990) Coherent qualitative probability. *J. Math. Psych.*, **34**:297–310.

[6] G. de Cooman (1997) Confidence Relations and Ordinal Information. *Inform. Sci.*, **104**:241–278.

[7] D. Dubois (1986) Belief Structure, Possibility Theory and Decomposable Confidence Measures on Finite Sets. *Comput. Artif. Intell.*, **5**(5):403–416.

[8] D. Dubois, H. Fargier, and H. Prade (1997) Decision-making under ordinal preferences and uncertainty. *AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pp. 41–46.

[9] D. Dubois, H. Prade, and R. Sabbadin (1997) A possibilistic logic machinery for qualitative decision. *AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pp. 47–54.

[10] D. Dubois, H. Fargier, and P. Perny (2003) Qualitative decision theory with preference relation and comparative uncertainty: An axiomatic approach. *Artificial Intelligence*, **148**:219–260.

[11] P. H. Giang, P. P. Shenoy (2001) A comparison of axiomatic approaches to qualitative decision making under possibility theory. In *Proc. of the* 17^{th} *Conference on Uncertainty in Artificial Intelligence UAI01*, pp. 162–170.

[12] M. Gelfond and V. Lifschitz (1988) *The stable model semantics for logic programming*. In *Proc. of 5th ILPS Conference*, pp. 1070–1080.

[13] M. Gelfond and V. Lifschitz (1991) *Classical negation in logic programs and disjunctive databases*, New Generation Computing, pp. 365–387.

[14] D. Lehmann (1996) Generalized qualitative probability: Savage revisited. In *Proc. of the* 12^{th} *Conference on Uncertainty in Artificial Intelligence UAI96*, pp. 381–388.

[15] V. Lifschitz (1999) *Answer Set Planning*. In D. De Schreye ed., Proc. of ICLP'99, The MIT Press, pp. 23–37.

[16] T. Syrjänen (2001) *Lparse 1.0 User's Manual*. Available at `www.tcs.hut.fi/Software/smodels`.

[17] W. Marek and M. Truszczyński (1999) *Stable models and an alternative logic programming paradigm*. The Logic Programming Paradigm: a 25-Year Perspective, Springer, pp. 375–398.

[18] P. Walley and T. Fine (1979) Varieties of modal (classificatory) and comparative probability. *Synthese*, **41**:321–374.

[19] S. K. M. Wong, Y. Y. Yao, P. Bollmann, and H. C. Bürger (1991) Axiomatization of Qualitative Belief Structure. *IEEE Trans. Systems Man Cybernet.*, **21**:726–734.

[20] Web references for some ASP solvers.
ASSAT: `assat.cs.ust.hk`
CCalc: `www.cs.utexas.edu/users/tag/cc`
Cmodels: `www.cs.utexas.edu/users/tag/cmodels`
DeReS: `ftp.cs.engr.uky.edu/cs/software/logic`
DLV: `www.dbai.tuwien.ac.at/proj/dlv`
Smodels: `www.tcs.hut.fi/Software/smodels`