A declarative approach to uncertainty orders*

Andrea Capotorti[1] and Andrea Formisano[2]

[1] Dipartimento di Matematica e Informatica, Università di Perugia.
`capot@dipmat.unipg.it`
[2] Dipartimento di Informatica, Università dell'Aquila.
`formisano@di.univaq.it`

Abstract. Traditionally, most of the proposed probabilistic models of decision under uncertainty rely on numerical measures and representations. Alternative proposals call for qualitative (non-numerical) treatment of uncertainty, based on preference relations and belief orders.
The automation of both numerical and non-numerical frameworks surely represents a preliminary step in the development of inference engines of intelligent agents, expert systems, and decision-support tools.
In this paper we exploit Answer Set Programming to formalize and reason about uncertainty expressed by belief orders. 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 orders, answer set programming, partial assessments, general inference.

Probability does not exist!
—Bruno de Finetti [13]

Introduction and background

Nowadays, several numerical tools are usually adopted in AI to represent and manage uncertainty. All of them originate from amendments of the well-known Probability measure, aimed at generalizing it to better fit different peculiarities of specific application fields (for a survey the reader can refer to [21, 30] or to [25, Chapters 8–10], among others. The Appendix briefly summarizes some basic notions about uncertainty measures, from the quantitative point of view). The measures that achieved wider diffusion can be classified as:

* Research partially funded by the *Information Society Technologies programme of the European Commission, Future and Emerging Technologies* under the IST-2001-37004 WASP project.

- Capacities;
- Possibility and Necessity measures;
- Probabilities;
- Belief and Plausibility functions;
- Lower and Upper probabilities.

Among all these measures (which are real-valued functions), Capacities [7] characterize the weakest notion. Indeed, Capacities are measures whose unique property is monotonicity with respect to the implication of events. Namely, if a situation A implies a situation B, then the uncertainty on A should be not greater than the uncertainty on B. Uncertainty models based on such measures are very general but, on the other hand, very weak because they describe nothing more than "common sense" behaviors. The class of Capacities includes all other classes.

Possibility measures (Necessity measures are their dual, cf. Def. 2 of the Appendix) come from *Fuzzy theory* [17, 33] and originate from the need to express "vagueness" about the descriptions of situations instead of uncertainty about their truth.

Probabilities are characterized by the "additivity" property: Having judged the uncertainties $P(A)$ and $P(B)$ on any pair of disjoint situations A and B, the uncertainty on their combination $A \vee B$ is defined as $P(A) + P(B)$. Such measures have a wide range of applications. Almost any medical, engineering, economic, and environmental decision-aid tool is usually built on (or at least compared to) probabilistic models.

Belief functions (whose dual are Plausibilities) are the base of *Evidence theory* [26]. The Belief on a proposition represents the "strength" by which a not fully detailed information supports its truth. Plausibility functions, on the other hand, represent how much the evidence makes reasonable that a proposition is true. Such uncertainty measures have found valuable application in economic and medical frameworks where the initial available information is quite not-specific.

Lower probabilities (whose dual are Upper probabilities) are instead adopted whenever one needs to consider as valid an entire family of probabilistic models in place of a single one. Such measures have been developed within the field of *Imprecise probabilities* [11, 29]. Obviously, such uncertainty measures are usually adopted in each context where precise probabilities are typically used, but where there are not enough constraints to be obliged to use a unique model.

As a matter of fact, each one of the framework described so far, can manage uncertainty and retains all of the expressive power of mathematical quantitative models. Though, inevitably, they suffer from the drawbacks often faced whenever numerical models are applied to practical problems: *a)* the difficulty of expressing a complete evaluation, and *b)* the hardness to elicit precise numerical values. The former problem can be circumvented by following the pioneering approach proposed by de Finetti in the context of Probabilities [12, 13]. Namely, by introducing the so called *partial models*, i.e. numerical evaluations defined only on some of the situations at hand, and intended to be a restriction of some of the complete models mentioned above. (Then, we will deal with partial Capacities, partial Probabilities, and so on.) This approach allows the analyst of the problem to focus his/her evaluation on the situations really judged relevant, w.r.t. the problem at hand. This leaves open the possibility to enlarge the model to other scenarios that could enter on the scene later. To obviate the latter drawback of numerical models, *qualitative approaches* have been proposed in the last decades. The central idea of such methodologies is to grade uncertainty about the truth of propositions, through comparisons expressing the judgement of "less or more believed to be true". This operationally translates into the use of (partial) order relations in place of numerical grades.

Qualitative approaches are receiving wider and wider attention, either as theoretical tools to deal directly with belief management [3, 10, 14], or inside the more articulated framework of decision-making theory (see, for example, [15, 16, 18, 20, 22]). This is because, they better fit the nature of human judgments.

Numerical models remain anyway a reference point. Both because their properties are well-known and deeply investigated, and because, when profitably involved, they could bring to conclusions hardly achievable by purely qualitative tools. The connection between the qualitative and the numerical frameworks is usually expressed by the requirement that the qualitative order must be representable[3] by a (partial) numerical model. Representability of an order guarantees that the comparisons among the propositions follow the same rationale of the kind of numerical model agreeing with. Hence, the basic properties of the way in which different pieces of information are combined is maintained.

In the next section we show that representability of orders, defined on arbitrary finite sets of propositions, can be characterized by the specific properties (axioms). Before to enter into such details, it is worth stressing that in this paper we adopt an alternative approach, by inverting the usual attitude towards qualitative management of uncertainty. In fact, specific axioms are usually set in advance, so that only order relations satisfying them are admitted. Here, on the contrary, given a fixed preference relation (for instance, directly issuing from analyst's interpretation of real world), our goal consists in ascertain what are the reasonable rules to work with. This will be made easy thanks to the expressive power of Answer Set Programming [23, 24]. In fact, most of such axioms are of direct declarative reading, as they involve only logical and preference relations. As we will see, such a declarative character supports a straightforward translation of the axioms within the logical framework of Answer Set Programming. As a consequence, we immediately obtain an executable specification able to discriminate between the different uncertainty orders. More specifically, we exploit a solver (in our case smodels, cf. [1]) to determine the set of axioms that are violated by a given preference relation, which expresses user's beliefs comparisons.

Then, we move the first step toward the implementation of an inference engine that borrows user's conceptualization of uncertainty and (implicitly) adopts

[3] Recall that, in general, a numerical assessment f on a set of propositions $A_1, \dots, A_n$ represents (or, equivalently, induces) a qualitative order $\preceq^*$ among them if, for each pair A_i, A_j it holds that $A_i \preceq^* A_j \iff f(A_i) \leqslant f(A_j)$.

his/her own way of modeling the intrinsic properties of the problem at hand. Thus, the system tries to mimic user's way of expressing lack of information and variability of phenomena. By acting in this manner, once the (most specific) framework closest to user's modelization is detected, it can be used to infer reasonable conclusions about proposition not comprised in the initial domain. This process is usually referred to as *order extension*. The availability of order-extension techniques is one of the main advantages offered by the use of partial models in the treatment of uncertainty.

The paper is organized as follows. Next section briefly describes the axioms characterizing partial uncertainty relations (notice that we focus on the treatment of partial orders, even if total relations can easily be dealt with by exploiting the very same machinery). Sec. 2 recalls the main features of Answer Set Programming, with particular emphasis on the application to the above mentioned issues. In Sections 3 and 4 we illustrate, also by simple examples, the potentialities of our approach. Finally, we draw conclusions and outline future developments.

1 Characterization of uncertainty orders

> *When one admits that nothing is certain one must, I think, also add that some things are more nearly certain than others.*
>
> —Bertrand Russell

By following the way paved by [8, 14, 31, 32], various (qualitative) preference orders have been fully classified in [4, 5, 6] according to their agreement with the most well-known numerical models; both for complete and partial assessments.

In particular, apart from Possibility and Necessity measures—that seem to have an intrinsically numerical character— [6] proposes a fully axiomatic classification of partial orders according to the numerical models outlined above.

Let us start by briefly recalling the basic notions on uncertainty orders and their axiomatic characterization. We will not enter into the details of the motivations for such classification, the reader is referred to [4, 5, 6]. The domain of discernment is represented by a finite set of events $\mathcal{E} = \{E_1, \ldots, E_n\}$ (among them, $\emptyset$ 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/her 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 reason, 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 predicating on 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, $\mathcal{A}_{\mathcal{E}}$ can be referenced as a supporting structure.

Let $\preceq$ be a partial (i.e. not necessarily defined for all pairs (A, B) in $\mathcal{E} \times \mathcal{E}$) order 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.

As mentioned before, Capacities constitute the most general numerical tool to manage uncertainty and they express "common sense" behaviors. Hence, in our context, any reasonable relation $\preceq$ must be representable by a partial Capacity (i.e., a restriction to the events under consideration, of a Capacity measure). This translates into the following axioms: the (partial) order $\preceq$ must be a reflexive binary relation on $\mathcal{E}$ such that

(A1) $\prec$ has no intransitive cycles;[4]

(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 belong to $\prec$.
Mathematical properties of orders satisfying basic axioms (A1), (A2) and (A3) are deeply investigated in [10]. In what follows, we consider these axioms as prerequisites for any investigation on $\preceq$. Differentiation among order relations can be done on the basis of more specific way of combining distinct pieces of information. Below, we list the axioms characterizing each class.[5] The name of the classes comes from the representability of $\preceq$ by corresponding partial numerical measures.[6]

Comparative Probabilities. An order $\preceq$ is *representable by a partial Probability assessment* iff the following holds:

(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 $r_1, \ldots, r_n > 0$, if $\sup \sum_{i=1}^{n} r_i(a_i - b_i) \leqslant 0$ holds than, for all $i = 1, \ldots, n$, $A_i \sim B_i$ (a_i, b_i denote the indicator functions of A_i, B_i, resp.).

Comparative Beliefs. An order $\preceq$ is *representable by a partial Belief function assessment* iff for all $A, B, C \in \mathcal{E}$ s.t. $A \subset B$, $B \wedge C = \phi$ it holds that

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

Comparative Lower probabilities. An order $\preceq$ is *representable by a partial Lower probability assessment* iff for all $A, B \in \mathcal{E}$ s.t. $A \wedge B = \phi$ it holds that

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

---

4 A preference relation $\prec$ on a set X has an intransitive cycle if there exist $A_1, \ldots, A_n \in X$ for $n > 2$ such that $A_i \prec A_{i+1}$ holds for each $i = 1, \ldots, n-1$, while $A_1 \prec A_n$ does not hold.

5 Note that we characterize each class by a single axiom, whereas in [6] some classes are described by introducing further axioms. It is easy to see that these additional axioms are redundant whenever we consider to enlarge $\prec$ by monotonicity (i.e. by imposing that $A \subseteq B \Longleftrightarrow A \prec B$ always holds).

6 Axiom (CP) was originally introduced in [8]. Axiom (B) derives by the analogous axiom introduced for complete orders in [32].

Comparative Plausibilities. An order $\preceq$ is *representable by a partial Plausibility function assessment* iff for all $A, B, C \in \mathcal{E}$ s.t. $A \subset B$ it holds that

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

Comparative Upper probabilities. An order $\preceq$ is *representable by a partial Upper-probability assessment* iff for all $A, B, C \in \mathcal{E}$ s.t. $A \wedge B = \phi$ it holds that

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

Comparative Lower/Upper probabilities. An order $\preceq$ can be simultaneously *represented by both a partial Lower-probability assessment* and *by a partial Upper-probability assessment* iff it simultaneously satisfies both axioms (L) and (U).

Note that only the axiom (CP) does not have a pure qualitative nature since it involves indicator functions and summations. Such axiom is the only one whose verification should require some form of numerical elaboration (e.g. involving some linear programming tool such as the simplex or the interior point methods). Meanwhile, to remain within the same kind 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 partial Probability assessment; nevertheless, its failure witnesses non-representability.

***Weak* comparative probabilities.** If $\preceq$ is *representable by a partial Probability assessment* then, for all $A, B, C \in \mathcal{E}$ s.t. $A \wedge C = B \wedge C = \phi$ it holds that

(WC) $\quad A \preceq B \Longrightarrow \neg(B \vee C \prec A \vee C)$

Clearly, all such 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.

The introduction of different classes of orders shares the very same motivations supporting the definition of different numerical measures of uncertainty. The main point is that there exist practical situations where a strictly probabilistic approach is not viable. The following example describes an extremely simplified situation of this kind.

Example 1. Let A, B, and C be three distinct companies, and let each of them be a potential buyer of a firm that some other company wants to sell. Even being distinct, both A and C belong to the same holding. Hence, the following uncertainty order about which company will be the buyer, could reflect specific information about the companies' strategies (by abuse of notation, let A denote the event "the company A buys the firm", and similarly for B and C):

$$\emptyset \prec A \prec B \prec B \vee C \prec A \vee C \prec \Omega.$$

Since A, B and C are incompatible events, it is immediate to see that the order relation is not representable by a probability because it violates axiom (WC), while it can be managed in line with Belief functions behaviors because it agrees with axiom (B).

2 Answer set programming

In the following sections we show how to obtain executable specifications from the axiomatic classification of preference orders described so far. To this end, we employ Answer Set Programming (ASP, for short).

Let us first briefly recall the basics of such alternative style of logic programming [23, 24]. A problem can be encoded—by using a function-free logic language—as a set of properties and constraints which describe the (candidate) solutions. More specifically, an *ASP-program* is a collection of *rules* of the form

$$L_1; \ldots; L_k; \mathit{not}\, L_{k+1}; \ldots; \mathit{not}\, L_\ell \;\leftarrow\; L_{\ell+1}, \ldots, L_m, \mathit{not}\, L_{m+1}, \ldots, \mathit{not}\, L_n$$

where $n \geqslant m \geqslant \ell \geqslant k \geqslant 0$ and each L_i is a literal, i.e., an atom A or a negation of an atom $\neg A$. The symbol $\neg$ denotes classical negation, while *not* stands for negation-as-failure (Notice that $','$ and $';'$ stand for logical conjunction and disjunction, respectively.) The left-hand side and the right-hand side of the clause are said *head* and *body*, respectively. A rule with empty head is a *constraint*. Intuitively, the literals in the body of a constraint cannot be all true, otherwise they would imply falsity.

Semantics of ASP is expressed in terms of *answer sets* (or equivalently *stable models*, cf. [19]). Consider first the case of an ASP-program P which does not involve negation-as-failure (i.e., $\ell = k$ and $n = m$). In this case, a set X of literals is said to be closed under P if for each rule in P, whenever $\{L_{\ell+1}, \ldots, L_m\} \subseteq X$, it holds that $\{L_1, \ldots, L_k\} \cap X \neq \emptyset$. If X is inclusion-minimal among the sets closed under P, then it is said to be an answer set for P. Such a definition is extended to any program P containing negation-as-failure by considering the *reduct* P^X (of P). P^X is defined as the set of rules

$$L_1; \ldots; L_k \;\leftarrow\; L_{\ell+1}, \ldots, L_m$$

for all rules of P such that X contains all the literals $L_{k+1}, \ldots, L_\ell$, but does not contain any of the literals $L_{m+1}, \ldots, L_n$. Clearly, P^X does not involve negation-as-failure. The set X is an answer set for P if it is an answer set for P^X.

Once a problem is described as an ASP-program P, its solutions (if any) are represented by the answer sets of P. Notice that an ASP-program may have none, one, or several answer sets.

Let us consider the program P consisting of the two rules

$$p; q \leftarrow \qquad\qquad \neg r \leftarrow p.$$

Such a program has two answer sets: $\{p, \neg r\}$ and $\{q\}$. If we add the rule (actually, a constraint) $\leftarrow q$ to P, then we rule-out the second of those answer sets, because it violates the constraint. This simple example reveals the core of the usual approach followed in formalizing/solving a problem with ASP. Intuitively speaking, the programmer adopts a "generate-and-test" strategy: first (s)he provides a set of rules describing the collection of (all) potential solutions. Then, the addition of a group of constraints rules-out all those answer sets that are not desired real solutions.

To find the solutions of an ASP-program, an ASP-solver is used. Several solvers have became available (cf. [1], for instance), each of them being characterized by its own prominent valuable features.

Expressive power of ASP, as well as, its computational complexity have been deeply investigated. The interested reader can refer to the survey [9], among others, for a comparison of expressive power and computational complexity of various forms of logic programming.

As we will see, in this work we choose smodels as solver, together with its natural front-end lparse [28].

Let us give a simple example of ASP-program (see [2], among others, for a presentation of ASP as a tool for declarative problem-solving). In doing this, we will recall the syntax of smodels as well as the main features of lparse/smodels which will be exploited in the rest of the paper (see [28], for a much detailed description). The problem we want to formalize in ASP is the well-known n-*queens* problem: "Given a $n \times n$ chess board, place n queens in such a way that no two of them attack each other". The clauses below state that a candidate solution is any disposition of the queens, provided that each column of the board contains one and only one queen. (The fact that a queen is placed on the n^{th} column and on the m^{th} row is encoded by the atom queen(n,m).)[7]

```
position(1..n).
1{queen(Col,Row) : position(Col)}1 :- position(Row).
```

The second rule is a particular form of constraint available in smodels' language. The general form of such a kind of clauses is

$$k\{\langle property\_def\rangle\text{:}\langle range\_def\rangle\}m \text{ :-}\langle search\_space\rangle$$

where: the conditions $\langle search\_space\rangle$ in the body define the set of objects of the domain to be checked; the atom $\langle property\_def\rangle$ in the head defines the property to be checked; the conjunction $\langle range\_def\rangle$ defines the possible values that the property may take on the objects defined in the body, namely by providing a conjunction of unary predicates each of them defining a range for one of the variables that occur in $\langle property\_def\rangle$ but not in $\langle search\_space\rangle$; k and m are the minimum and maximum number of values that the specified property may take on the specified objects. (Notice that this form of constraint, available in smodels, actually is syntactic sugar, since it can be translated into "proper" ASP-clauses thanks to negation, cf. [28, 27].)

We now introduce two constraints, in order to rule out those placements where two queens control either the same row or the same diagonal of the board:

```
:- queen(Col,Row1), queen(Col,Row2),
   position(Col), position(Row1), position(Row2),
   Row1 < Row2.
:- queen(Col1,Row1), queen(Col2,Row2),
   position(Col1), position(Col2), position(Row1), position(Row2),
   Row1 < Row2, abs(Col1-Col2) == abs(Row1-Row2).
```

Here is some of the answer sets produced by smodels, when fed with our program (together with a value for the constant n, in this case we put n= 8).

[7] In the syntax of smodels ':-' denotes implication $\leftarrow$, while ',' stands for conjunction. Moreover, the constant n occurring in the first clause, can be seen as a parameter of the program, supplied to the solver at run-time.

```
Answer: 1.
Stable Model: queen(4,1) queen(6,2) queen(1,3) queen(5,4) queen(2,5)
  queen(8,6) queen(3,7) queen(7,8) ...
Answer: 2.
Stable Model: queen(4,1) queen(2,2) queen(8,3) queen(5,4) queen(7,5)
  queen(1,6) queen(3,7) queen(6,8) ...
...
```

Notice that lparse offers some elementary built-in arithmetic functions (such as abs(), in the above clause) that can be used to perform simple arithmetics. More in general, lparse allows the user to employ user-defined C or C++ functions within an ASP-program. The object code of these functions needs only to be linked with lparse at run time. (The interested reader is referred to [28] for a detailed description of this feature.) We exploited this feature (not directly available in some other solvers) to implement a basic library of functions aimed at handling sets and operation on sets.

The pair lparse/smodels constitutes an essential and neat tool for fast prototypical development. Moreover notable facilities come from the simple albeit useful capability of integration with the C programming language, the prompt availability of the source-code (under the GNU General Public License) and documentation, and the ease of use.

3 Preference classification

Our first task consists in writing an ASP-program able to classify any given partial order $\preceq$, 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(·).[8] Auxiliary predicates/functions are defined to render usual set-theoretical constructors, such as $\cap$, $\cup$, and $\subseteq$, which, as mentioned, have been made available by linking user-defined C-libraries.

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).
```

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

This 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, in this phase, we do not want to impose such axiom, but we just want to test whether or not it is satisfied by the preference relation at hand, we introduce a rule of the form:

```
failsB :- 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 failsB is true (i.e. belongs to the answer set) whenever there exist events falsifying axiom (B). Having in mind the axiom (B) of Sec. 1, this clause is of immediate reading. Analogous treatment has been done for all other axioms (L), (U), (PL), and (WC).

When smodels is fed with such program, together with a description of an input preference relation (i.e., a collection of facts of the forms prec(·,·), precneq(·,·), and equiv(·,·)), different outcomes may be obtained:

a) If no answer set is produced, then the input preference relation violates some basic requirement, such as axioms (A1), (A2), or (A3).

b) Otherwise, if an answer set is generated, there exists a numerical (partial) model representing the input preference order. Moreover, the presence in the answer set of a fact of the form fails$\mathcal{C}$ (say failsL, for example), witnesses that the corresponding axiom ((L) in the case) is violated by the given preference order. Consequently, the given order (as well as its extensions) is not compatible with the uncertainty framework ruled by $\mathcal{C}$ (in the case of failsL, the given order cannot be represented by a partial Lower probability).

Example 2. Suppose a physician wants to perform a preliminary evaluation about the reliability of a test for SARS (Severe Acute Respiratory Syndrome). Up to his/her 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). The elements appearing in his/her analysis can be schematized as:

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:

$$\mathsf{A}\cap\mathsf{B}=\emptyset,\ \ \mathsf{B}\cap\mathsf{C}=\emptyset,\ \ \mathsf{A}\cap\mathsf{C}=\emptyset,\ \ \mathsf{A}\cup\mathsf{B}\cup\mathsf{C}=\Omega,\ \ \mathsf{D}=\mathsf{A}\cup\mathsf{B},\ \ \mathsf{E}\subset\mathsf{C},\ \ \mathsf{F}\cap\mathsf{A}=\emptyset.$$

Consider the following partial order:

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).

Due to events' meaning, such order seems reasonable. If it is given as input to smodels, the answer set found includes the facts failsB and failsWC. This means that the given preference relation agrees with the basic axioms, however it cannot be managed by using neither a Probability nor a Belief function. Nevertheless, one can use comparative Lower probabilities or comparative Plausibilities. ■

4 Partial-order extension

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 extension retains the same character of the initial order (e.g., both satisfy the same axioms).

More precisely, let be given an initial (partial) assessment expressed as a set of known events $\mathcal{E}$ together with a (partial) order $\preceq$ over $\mathcal{E}$. Moreover, assume that $\preceq$ satisfies the axioms characterizing a specific class, say $\mathcal{C}$, of orders (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 [8, 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 $\mathcal{C}$. 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 $\mathcal{C}$.

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:

$$\begin{aligned} E \in \mathcal{L}_{\mathsf{S}} &\text{ iff no extension } \preceq^* \text{ of } \preceq \text{ can infer that } \mathsf{S} \preceq^* E \\ E \in \mathcal{WL}_{\mathsf{S}} &\text{ iff no extension } \preceq^* \text{ of } \preceq \text{ can infer that } \mathsf{S} \prec^* E \\ E \in \mathcal{U}_{\mathsf{S}} &\text{ iff no extension } \preceq^* \text{ of } \preceq \text{ can infer that } E \preceq^* \mathsf{S} \\ E \in \mathcal{WU}_{\mathsf{S}} &\text{ iff no extension } \preceq^* \text{ of } \preceq \text{ can infer that } E \prec^* \mathsf{S} \end{aligned}$$

Consequently, any order $\preceq^+$ extending $\preceq$ must, at least, impose that:

$$\begin{array}{ll} E \prec^+ \mathsf{S} \text{ for each } E \in \mathcal{L}_{\mathsf{S}}, & E \preceq^+ \mathsf{S} \text{ for each } E \in \mathcal{WL}_{\mathsf{S}}, \\ \mathsf{S} \prec^+ E \text{ for each } E \in \mathcal{U}_{\mathsf{S}}, \text{ and} & \mathsf{S} \preceq^+ E \text{ for each } E \in \mathcal{WU}_{\mathsf{S}}, \end{array}$$

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

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(s), 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 3, 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):

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

Rules of this kind (actually, constraints, in the sense described in Sec. 2), declare "undesirable" any extension for which the axiom is violated. For instance, consider a ground instance of the above rule; whenever the fact holdsL is present (i.e. is true in an answer set), then to make the (ground) clause 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 the fact holdsL to the input of the solver.

A further rule describes 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). Then, smodels produces as output the answer sets fulfilling the desired requirements and encoding "legal" total orders.

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 be 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. [1]). In general, a simple inspection of the answer sets generated by smodels allows one to detect which is the minimal extension of the preference relation which is mandatory for each total order.

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

Example 3. Consider the partial order of Example 2 and the new event:

$$\mathsf{S} \equiv \textit{The real state of having SARS}$$

subject to these restrictions: $\mathsf{S}\subset\mathsf{F}$ and $\mathsf{F}\cap\mathsf{E}\subset\mathsf{S}$. Since in Example 2 we discovered that the initial preference relation satisfies axiom (PL), we want to impose such axiom and compute the extension of the initial order.

Once filtered smodels' output, we obtained the following result:[9]

$$\begin{array}{llll} \mathsf{precneq(S,A\cup C)} & \mathsf{precneq(S,A\cup E)} & \mathsf{precneq(S,D)} & \\ \mathsf{precneq(S,A)} & \mathsf{precneq(S,}\Omega\mathsf{)} & \mathsf{prec(\emptyset,S)} & \mathsf{prec(S,F)} \end{array}$$

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. ■

The availability of automated tools able to extend preference orders, whenever new knowledge (new events) is acquired, directly suggests applications in expert systems and decision-support tools. In automated diagnosis, planning, or problem solving, to mention some examples, one could easily imagine scenarios where knowledge is not entirely available from the beginning. We could outline how a rudimental inference process could develop, by identifying the basic steps an automated agent should perform:

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

0) Acquisition of an initial collection of observations (events) about the object of the analysis, together with a (qualitative) partial preference assessment;
1) Detection of which is the most adequate (i.e., the most discriminant) uncertainty framework, through a "preference classification phase" (cf. Sec. 3);
2) Whenever new knowledge becomes available, refine agent's description of the real world by performing order extension (which substantially corresponds to knowledge inference. Cf. Sec. 4).

The results of step 2) could be then exploited to guide further investigations on the real world, in order to obtain new information. Then, step 2) will be repeated and the process will continue until further pieces of knowledge are obtainable or an enough accurate degree of believe is achieved.

Conclusions

In this paper we started an exploration of the potentialities offered by Answer Set Programming for building decision support systems based on qualitative judgments. Thanks to the remarkable features of ASP, the implementation of what could be thought as a kernel of an inference engine, sprouted almost naturally. Certainly, our research is at an initial stage and the implementation we reported on in this paper cannot be considered to be prototype. Next step in this research would consist in validating the proposed approach by means of a number of benchmarks aimed at testing our prototype on the ground of real applications. A comparison of its behaviour w.r.t. other possible declarative approaches, for instance exploiting Constraint Logic Programming, is due. Results of this activity will help in consolidating the prototype. In this context, a further goal consists in completing our approach so as to handle comparative Probabilities too. Since no axiomatic characterization of comparative Probabilities is known (up to our knowledge), this aim should be achieved through integration with efficient linear optimization tools (such as the column generation techniques). More in general, we envisage the design of a full-blown automated system which integrates different (in someway complementary) techniques and methods for uncertainty management; comprehending mixed numerical/qualitative assessments and conditional frameworks.

Acknowledgments

We thank the anonymous referees for the useful remarks, as well as Stefania Costantini and Agostino Dovier for fruitful discussions on the topics of this paper.

References

[1] Web references for some ASP solvers.
ASSAT: `http://assat.cs.ust.hk`
CCalc: `http://www.cs.utexas.edu/users/tag/cc`

Cmodels: `http://www.cs.utexas.edu/users/tag/cmodels`
DeReS: `ftp://ftp.cs.engr.uky.edu/cs/software/logic`
DLV: `http://www.dbai.tuwien.ac.at/proj/dlv`
Smodels: `http://www.tcs.hut.fi/Software/smodels`.

[2] C. Baral. *Knowledge representation, reasoning and declarative problem solving.* Cambridge University Press, 2003.

[3] T. Bilgiç. Fusing interval preferences. In *Proc. of EUROFUSE Workshop on Preference Modelling and Applications*, pages 253–258, 2001.

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

[5] A. Capotorti and B. Vantaggi. Relationships among ordinal relations on a finite set of events. In B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, editors, *Information, Uncertainty, Fusion*, pages 447–458. Kluwer, 1998.

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

[7] G. Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5:131–295, 1954.

[8] G. Coletti. Coherent qualitative probability. *J. Math. Psych.*, 34(3):297–310, 1990.

[9] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[10] G. de Cooman. Confidence relations and ordinal information. *Inform. Sci.*, 104:241–278, 1997.

[11] G. de Cooman and P. Walley. The imprecise probability project. Available at :`http://ippserv.rug.ac.be/`.

[12] B. de Finetti. Sul significato soggettivo della probabilità. *Fundamenta Mathematicae*, 17:298–321, 1931. Engl. transl. in P. Monari and D. Cocchi, editors, *Induction and Probability*, CLUEB, Bologna: 291–321, 1993.

[13] B. de Finetti. *Theory of Probability.* Wiley, 1974. (Italian original *Teoria della probabilità*, Einaudi, Torino, 1970).

[14] D. Dubois. Belief structure, possibility theory and decomposable confidence measures on finite sets. *Comput. Artif. Intell.*, 5:403–416, 1986.

[15] D. Dubois, H. Fargier, and P. Perny. Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach. *Artif. Intel.*, 148:219–260, 2003.

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

[17] D. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications.* Academic Press, New York, 1980.

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

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

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

[21] G. J. Klir and T. A. Folger. *Fuzzy sets, uncertainty, and information.* Prentice-Hall, 1988.

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

[23] V. Lifschitz. Answer set planning. In D. De Schreye, editor, *Proc. of ICLP'99*, pages 23–37. The MIT Press, 1999.
[24] W. Marek and M. Truszczyński. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-Year Perspective*, pages 375–398. Springer, 1999.
[25] H. T. Nguyen and E. A. Walker. *A first course in fuzzy logic.* CRC Press, Boca Raton, 1997.
[26] G. Shafer. *A mathematical theory of evidence.* University of Princeton Press, Princeton, 1976.
[27] P. Simons. *Extending the Smodels System with Cardinality and Weight Constraints.* PhD thesis, Helsinki University, Department of Computer Science and Engineering, July 2000. Research report 58 of Technology Laboratory for Computer Science, HUT-TCS-A58.
[28] T. Syrjänen. Lparse 1.0 user's manual, 1999. Available at `http://www.tcs.hut.fi/Software/smodels`.
[29] P. Walley. *Statistical reasoning with imprecise probabilities.* Chapman and Hall, London, 1991.
[30] P. Walley. Measures of uncertainty in expert systems. *Artif. Intel.*, 83:1–58, 1996.
[31] P. Walley and T. Fine. Varieties of modal (classificatory) and comparative probability. *Synthese*, 41:321–374, 1979.
[32] S. K. M. Wong, Y. Y. Yao, P. Bollmann, and H. C. Bürger. Axiomatization of qualitative belief structure. *IEEE Trans. Systems Man Cybernet.*, 21:726–734, 1991.
[33] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

A gentle introduction to uncertainty measures

In this appendix we briefly describe the various generalizations of Probability measures used in this paper, as introduced in standard literature. The following material is far from being an exhaustive and complete treatment. We will give just a informal introduction to the subject. The interested reader can refer to the widely available literature. Introductory treatment of the relationships between Probability measures, belief functions and possibility measures, can be found in [21, 25, 30], to mention some among many.

We will consider, as domain of interest, a set Ω of possibilities (Ω is often refered to as *sample space*). For our purposes it is sufficient to consider the case of a finite domain. An *event* is then defined as a subset of Ω. In order to introduce uncertainty measures, we can consider any algebra $\mathcal{A}$ (on Ω), consisting of a set of subsets of Ω, such that $\Omega \in \mathcal{A}$, and closed under union and complementation.

All of the measures we are going to introduce will be (normalized) monotone real-valued functions over an algebra. Such functions are usually called (*Choquet*) *Capacities* [7], even if they are refered also as *fuzzy measures* or *Sugeno measures.*

Definition 1. *A real-valued function F on 2^{Ω} is a* Capacity *if it holds that*
$F(\emptyset) = 0$, $\quad F(\Omega) = 1$, $\quad$ *and for all* $A, B \subseteq \Omega \quad A \subseteq B \Longrightarrow F(A) \leqslant F(B)$.

Let us denote the class of Capacities over Ω by $CAP(\Omega)$.

The notion of Capacity is often too general to be of interest by itself. In fact adopting it, apart from monotonicity, there is no other relationship imposed between the uncertainty assigned to a composed event, e.g. $F(A \cup B)$, and the uncertainty of its components $F(A)$ and $F(B)$. In order to reflect different rationales in managing the information, several constraints can be imposed on the manner in which uncertainties of composed events are determined. In what follows we describe some of the more interesting measures obtained by imposing further conditions on measures, apart to be a Capacity. We start with the most adopted measure of uncertainty. It is characterized by the additivity property of combination: A *Probability* P over Ω is a capacity which satisfies the following *additivity* requirement: For all $A, B \subseteq \Omega$ with $A \cap B = \emptyset$ $P(A \cup B) = P(A) + P(B)$. The class of all Probabilities over Ω is denoted by $PROB(\Omega)$. Clearly, we have that $PROB(\Omega) \subseteq CAP(\Omega)$. Let us introduce a further concept:

Definition 2. *Let F_1 and F_2 be two functions on 2^{Ω}. Then, F_1 is the* dual *of F_2 if for each $A \subseteq \Omega$ it holds that $F_1(A) = 1 - F_2(\Omega \setminus A)$.*

Note that the dual of a Capacity is a Capacity too. Moreover, the dual of a Probability is the Probability itself.

Additivity, even being widely adopted in “measurement” processes, is usually thought to be a too strong requirement. Hence, several generalizations have been proposed. In particular, the following definition characterizes those Capacities satisfying only one of the weak inequalities which, taken together, give additivity.

Definition 3. *Let Π and N be Capacities over Ω.*

- *Π is a* Possibility *measure (over Ω) if it satisfies the following property: For all $A, B \subseteq \Omega$ $\Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\}$.*
- *N is a* Necessity *measure (over Ω) if it is the dual of a Possibility measure.*

It is immediate to see that

- a Possibility measure Π satisfies the *sub-additivity* property:
  For all $A, B \subseteq \Omega$ $\Pi(A \cup B) \leqslant \Pi(A) + \Pi(B)$;
- A Necessity measure N satisfies the *super-additivity* property:
  For all $A, B \subseteq \Omega$ $N(A \cup B) \geqslant N(A) + N(B)$.

A Possibility measure Π is usually induced by a *possibility distribution* (i.e. a fuzzy set) $\pi : \Omega \to [0, 1]$. The value $\pi(x)$ expresses the possibility of a singleton $x \in \Omega$ to be representative of the concept being considered. Possibility is then defined by putting $\Pi(A) = \max\{\pi(x) \mid x \in A\}$ for any $A \subseteq \Omega$. The classes of Possibilities and Necessities over Ω are denoted by $POS(\Omega)$ and $NEC(\Omega)$, respectively.

Let us consider now a slightly different situation. Suppose that the available (possibly incomplete) knowledge permits the formulation of some form of constraint on the Probability of the events. Ideally, such constraints may determine a unique Probability measure. In general, this is not the case. In fact, there may be a non-void set of Probability measures which satisfy the given constraints. Here we describe the measures induced by such set. In particular, a set of Probabilities measures (over Ω) induces two natural measures. Namely, its lower and upper envelope.

Definition 4. *Let $\emptyset \neq \mathcal{P} \subseteq PROB(\Omega)$. The* lower envelop $\underline{\mathcal{P}}$ *and the* upper envelopes $\overline{\mathcal{P}}$ *of $\mathcal{P}$ are defined as:*

- *For each $A \subseteq \Omega$, $\underline{\mathcal{P}}(A) = inf\{P(A) \mid P \in \mathcal{P}\}$;*
- *For each $A \subseteq \Omega$, $\overline{\mathcal{P}}(A) = sup\{P(A) \mid P \in \mathcal{P}\}$.*

Lower envelopes are usually called Lower probability measures*, while upper envelopes, which are their duals, are called* Upper probability measures*.*

Let us denote the classes of Lower and Upper probabilities over Ω by $LOWP(\Omega)$ and $UPP(\Omega)$, respectively.

It remains to introduce Belief and Plausibility measures. With the most general formulation, following [26], we have:

Definition 5. *A function $Bel : 2^{\Omega} \to [0,1]$ is a* Belief *measure if it is a Capacity and it satisfies the following condition (known as* ∞-monotonicity*).*

For each $n \geqslant 1$, $Bel\big(\bigcup_{i=1}^{n} A_i\big) \geqslant \sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} Bel\big(\bigcap_{i \in I} A_i\big)$

(where $A_i \subseteq \Omega$ for each i).

Intuitively speaking, a Belief function Bel is usually constructed through a basic assignment of uncertainty, not necessarily being a Capacity, $\mu : 2^{\Omega} \to [0,1]$ so that, for any proposition $A \subseteq \Omega$, $Bel(A) = \sum_{B \subseteq A} \mu(B)$.

Notice that Belief functions are often called *Capacities monotone of infinite order*. Capacities which satisfy the above condition with the restriction that $n \leqslant N$ are then said *monotone of order* N (or N-monotone). Dually, if the opposite inequality ($\leqslant$) is considered, the measure is said to be an N*-alternating* capacity. For $N = 2$ these properties reduce to usual super- and sub-additivity, respectively.

The dual of a Belief measure is called *Plausibility* measure. The classes of Belief measures and Plausibility measures are denoted by $BEL(\Omega)$ and $PL(\Omega)$, respectively.

The following relationships can be shown to hold between the classes of Capacities seen so far:

$$CAP(\Omega) \supset LOWP(\Omega) \supset BEL(\Omega) \supset NEC(\Omega)$$
$$CAP(\Omega) \supset UPP(\Omega) \supset PL(\Omega) \supset POS(\Omega)$$
$$BEL(\Omega) \cap PL(\Omega) \supset PROB(\Omega).$$