Argumentation-Based Preference Modelling with Incomplete Information

Wietske Visser, Koen V. Hindriks and Catholijn M. Jonker

Man Machine Interaction Group, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
Wietske.Visser@tudelft.nl, K.V.Hindriks@tudelft.nl, C.M.Jonker@tudelft.nl

Abstract. No intelligent decision support system functions even remotely without knowing the preferences of the user. A major problem is that the way average users think about and formulate their preferences does not match the utility-based quantitative frameworks currently used in decision support systems. For the average user qualitative models are a better fit. This paper presents an argumentation-based framework for the modelling of and automated reasoning about multi-issue preferences of a qualitative nature. The framework presents preferences according to the lexicographic ordering that is well-understood by humans. The main contribution of the paper is that it shows how to reason about preferences when only incomplete information is available. An adequate strategy is proposed that allows reasoning with incomplete information and it is shown how to incorporate this strategy into the argumentation-based framework for modelling preferences.

Key words: Qualitative Preferences, Argumentation, Incomplete Information

1 Introduction

In this paper we introduce an argumentation-based framework for modelling qualitative multi-attribute preferences under incomplete information. This is motivated by our interest in developing a negotiation support system, as part of a larger project. In this context, we are faced with the need to express a user's preferences. A necessary (but not sufficient) condition for an offer to become an agreement is that both parties feel that it satisfies their preferences well enough. Unfortunately, eliciting and representing a user's preferences is not unproblematic. Existing negotiation support systems are based on quantitative models of preferences. These kinds of models are based on utilities; a utility function determines for each outcome a numerical value of utility. However, it is difficult to elicit such models from users, since humans generally express their preferences in a more qualitative way. We say we like something more than something else, but it seems strange to express liking something exactly twice as much as an alternative. In this respect, qualitative preference models will have a higher cognitive plausibility as they provide a better correspondence with representations used by humans. We also think that qualitative models will allow a human user to interact more naturally with an agent negotiating on his behalf or supporting him in his negotations, and will investigate this in future. There are, however, several challenges that need to be met before qualitative models can be usefully applied. Doyle and Thomason [8] provide an overview including among others the challenge to deal with partial information (information-limited rationality) and, more generally, the challenge to formalize various reasoning-related tasks (knowledge representation, reasons, and preference revision).

For any real-life application it is important to be able to handle multi-issue preferences. It is a natural approach to derive object preferences from general preferences over properties or attributes. For example, it is quite natural to say that you prefer one house over another because it is bigger and generally you prefer larger houses over smaller ones. This might still be so if the first house is more expensive and you generally prefer cheaper options. So there is an interplay between attributes and the preferences a user holds over them in determining object preferences. This means that object preferences can be quite complex. One approach to obtain preferences about objects is to start with a set of properties of these objects and derive preferences from a ranking of these properties that indicates the relative importance or priority of each of these properties. This approach to obtain preferences is typical in multi-attribute decision theory [12], a quantitative theory that derives object preferences from utility values assigned to outcomes which are derived from numeric weights associated with properties or attributes of objects. Several qualitative approaches have also been proposed [3, 5–7, 13].

A user's preferences and knowledge about the world may also be incomplete, inconsistent or changing. For example, a user may lack some information regarding the objects he has to choose between, or he might have contradictory information from different sources. Preferences may change for various reasons, e.g. new information becoming available, experience, changing goals, or interaction with persuasive others. For now, we focus on the situation in which information about objects is not complete, but will address other types of incompleteness, inconsistency and change in future.

The approach we take is based on argumentation. In recent years, argumentation has evolved to be a core study within artificial intelligence and has been applied in a range of different topics [2]. We incorporate some of the ideas introduced in existing qualitative approaches but also go beyond these approaches by introducing a framework that is able to reason about preferences also when only incomplete information is available. Because of its non-monotonic nature, argumentation is useful for handling inconsistent and incomplete information. Although a lot of work has been done on argumentation-based negotiation (for a comprehensive review, see [16]), most of this work considers only the bidding phase in which offers are exchanged. For preparation, the preferences of a user have to be made clear (both to the user himself and to the agent supporting him), hence we need to express and reason with them. We focus here on the modelling of a single user's preferences by means of an argumentation process. The idea is that a user weighs his preferences, which gives him better insight into his own preferences, and so this weighing is part of the preference elicitation process. The weighing of arguments maps nicely onto argumentation. For example, 'I like to travel by car because it is faster than going by bike' is countered by 'But cycling is healthier than driving the car and that is more important to me, so I prefer to take the bike'. This possibility to construct arguments that are attacked by counterarguments is another advantage of argumentation, since it is a very natural way of reasoning for humans and fits in with a user's own reasoning processes. This is a general feature of argumentation and we will make extensive use of it: arguments like those above form the basis of our system. We believe that this way of reasoning will also be very useful in the preference elicitation process since the user's insight into his preferences grows piece by piece as he is expressing them. The introduction of an argumentation-based framework for reasoning about preferences even when only incomplete information is available seems particularly suitable for such a step-by-step process. It allows the user to extend and refine the system representation of his preferences gradually and as the user sees fit. Another motivation to use argumentation is the link with multi-agent dialogues [1], which will be very interesting in our further work on negotiation.

In this paper we present an argumentation-based framework for reasoning with qualitative multi-attribute preferences. In Section 2, we introduce qualitative multi-attribute preferences, in particular the lexicographic preference ordering. In Section 3 we start by modelling this ordering for reasoning with complete information in an argumentation framework. Then we proceed and extend this framework in such a way that it can also handle incomplete information. Our main contribution, in Section 4, is a strategy (based on the lexicographic ordering) with some desired properties to derive object preferences in the case of incomplete information. In Section 5 this strategy is subsequently incorporated into the argumentation framework. Section 6 concludes the paper.

2 Qualitative Multi-Attribute Preferences

Qualitative multi-attribute preferences over objects are based on a set of relevant attributes or goals, which are ranked according to their importance or priority. Without loss of generality, we only consider binary (Boolean) attributes (cf. [5]). Moreover, it is assumed that the presence of an attribute is preferred over its absence. For example, given that *garden* is an attribute, a house that has a garden is preferred over one that does not have one. The importance ranking of attributes is defined by a total preorder (a total, reflexive and transitive relation), which we will denote by $\succeq$. This relation is not required to be antisymmetric, so two or more attributes can have the same importance. The relation $\succeq$ yields a stratification of the set of attributes into importance levels. Each importance level consists of attributes that are deemed equally important. Together with factual information about which objects have which attributes, the attribute ranking forms the basis on which various object preference orderings can be defined. One of the most well-known preference orderings is the lexicographic ordering, which we will use here. [5] and [7] define more multi-attribute preference orderings, such as the discrimin and best-out orderings. In this paper we focus on the lexicographic ordering because it seems natural, it defines a total preference relation (contrary to the discrimin ordering) and it is more discriminating than the best-out ordering. Since the other orderings are structurally similar to the lexicographic ordering, a similar argumentation framework could be defined for them if desired. We introduce the lexicographic preference ordering by means of an example.

Example 1. Paul wants to buy a house. According to him, the most important attributes are *large* (minimally 100m^2), *garden* and *closeToWork*, which among themselves are equally important. The next most important attributes are *nearShops* and *quiet*. Being *detached* is the least important. Paul can choose between three options: a *villa*,

	large	*garden*	*closeToWork*	*nearShops*	*quiet*	*detached*
villa	✓	✓				✓
apartment	✓		✓	✓		
cottage		✓		✓	✓	✓

Table 1. An example of objects and attributes

an *apartment* and a *cottage*. The attributes of these objects are displayed in Table 1. In this table, the attributes are ordered in decreasing importance from left to right. A dashed line between attributes indicates equal importance, a solid line a transistion to a lower importance level. A checkmark indicates that an object has the attribute, an empty box means that the attribute is absent. Which house should Paul choose? He first considers the highest importance level, which in this case comprises *large*, *garden* and *closeToWork*. The *villa* and the *apartment* both satisfy two of these attributes, while the *cottage* only satisfies one. So at this moment Paul concludes that both the *villa* and the *apartment* are preferred to the *cottage*. For the preference between the *villa* and the *apartment* he has to look further. At the next importance level, the *apartment* satisfies one attribute and the *apartment* satisfies none. So the *apartment* is preferred over the *villa*. Note that although the *cottage* satisfies the most attributes in total, it is still the least preferred option because of its bad score at the more important attributes.

Definition 1. (Lexicographic preference ordering) *Let $\mathscr{P}$ be a set of attributes or goals, and $\succeq$ a total preorder on $\mathscr{P}$. We write $P \succ Q$ for $P \succeq Q$ and $Q \not\succeq P$, and $P \approx Q$ for $P \succeq Q$ and $Q \succeq P$. We use $|\cdot|$ to denote the cardinality of a set. Object a is* strictly preferred *over object b according to the lexicographic ordering if there exists an attribute P such that $|\{P' \mid a \text{ satisfies } P' \text{ and } P \approx P'\}| > |\{P' \mid b \text{ satisfies } P' \text{ and } P \approx P'\}|$ and for all $Q \succ P$: $|\{Q' \mid a \text{ satisfies } Q' \text{ and } Q \approx Q'\}| = |\{Q' \mid b \text{ satisfies } Q' \text{ and } Q \approx Q'\}|$. Object a is* equally preferred *as object b according to the lexicographic ordering if for all P: $|\{P' \mid a \text{ satisfies } P' \text{ and } P \approx P'\}| = |\{P' \mid b \text{ satisfies } P' \text{ and } P \approx P'\}|$.*

3 Argumentation Framework for Complete Information

In order to formally model and reason with preferences we define an argumentation framework (AF). We use as our starting point the well-known argumentation theory of Dung [10]. An abstract AF in the sense of Dung consists of a set of arguments and a defeat relation (informally, a counterargument relation) among those arguments. An AF is abstract in the sense that both the set of arguments and the defeat relation are assumed to be given, and the construction and internal structure of arguments is not taken into account. If we want to reason with argumentation, we have to instantiate an abstract AF by specifying the structure of arguments and the defeat relation. Arguments are typically built from a logical *language* by chaining inferences. *Inferences* are instantiations of general inference schemes, such as modus ponens. *Defeat* is based on certain relations between the elements of arguments. Together with a knowledge base, they provide a specific AF for arguing about multi-attribute preferences.

3.1 Language

The language has to allow us to express everything we want to talk about when reasoning about preferences. To start, we need to be able to state the facts about objects: which attributes they do and do not have. We also have to express the importance ranking of attributes, so we need to be able to say that one attribute is more important than another, or that two attributes are equally important. Of course, we want to say that one object is preferred over another, and that two objects are equally preferred. Finally, we need to be able to express how many attributes of equal importance a certain object has, since the lexicographic preference ordering is based on counting these. To this end, we introduce a special predicate $has(a,[P],n)$ which expresses that object a has n attributes of the importance level of attribute P. Since we have no names for importance levels, we denote them by any attribute of that level, placed between square brackets. It is not necessary that the attribute used is among the attributes that the object has; in our example, $has(apartment,[quiet],1)$ is true even though the *apartment* is not *quiet*. All of the things described can be expressed in the following language.

Definition 2. **(Language)** *Let $\mathscr{P}$ be a set of attribute names with typical elements P,Q, and $\mathscr{O}$ a set of object names with typical elements a,b, and let n be a non-negative integer. The* language $\mathscr{L}$ *is defined as follows.*

$$\varphi \in \mathscr{L} ::= P(a) \mid P \succ Q \mid P \approx Q \mid pref(a,b) \mid eqpref(a,b) \mid has(a,[P],n) \mid \neg\varphi$$

Formulas of this language have the following informal meaning:

$P(a)$	object a has attribute P
$P \succ Q$	attribute P is more important than attribute Q
$P \approx Q$	attribute P is equally important as attribute Q
$pref(a,b)$	object a is strictly preferred over object b
$eqpref(a,b)$	object a is equally preferred as object b
$has(a,[P],n)$	object a has n attributes equally important as attribute P (not necessarily including P itself)
$\neg\varphi$	the negation of φ

The idea is that preferences over objects are derived from facts about which objects have which attributes, and the importance order among attributes. These facts are contained in a *knowledge base*, which is a set of formulas of the type $P(a)$, $\neg P(a)$, $P \succ Q$ and $P \approx Q$. A knowledge base is complete if, given a set of objects to compare and a set of attributes to compare them on, it contains for every object a and for every attribute P, either $P(a)$ or $\neg P(a)$, and for all attributes P,Q, either $P \succ Q$, $Q \succ P$ or $P \approx Q$.

Example 2. The information from Example 1 can be expressed in the form of the following knowledge base that is based on the language $\mathscr{L}$.

$large \approx garden \approx closeToWork \succ nearShops \approx quiet \succ detached$

$large(villa)$	$large(apartment)$	$\neg large(cottage)$
$garden(villa)$	$\neg garden(apartment)$	$garden(cottage)$
$\neg closeToWork(villa)$	$closeToWork(apartment)$	$\neg closeToWork(cottage)$
$\neg nearShops(villa)$	$nearShops(apartment)$	$nearShops(cottage)$
$\neg quiet(villa)$	$\neg quiet(apartment)$	$quiet(cottage)$
$detached(villa)$	$\neg detached(apartment)$	$detached(cottage)$

1	$\dfrac{}{has(a,[P],0)}$	$count(a,[P],\varnothing)$
2	$\dfrac{P_1(a) \quad \dots \quad P_n(a) \quad P_1 \approx \dots \approx P_n}{has(a,[P_1],n)}$	$count(a,[P_1],\{P_1,\dots P_n\})$
3	$\dfrac{P_1(a) \quad \dots \quad P_n(a) \quad P_1 \approx \dots \approx P_n}{count(a,[P_1],S \subset \{P_1,\dots,P_n\}) \text{ is inapplicable}}$	$count(a,[P_1],\{P_1,\dots,P_n\})uc$
4	$\dfrac{has(a,[P],n) \quad has(b,[P'],m) \quad P \approx P' \quad n > m}{pref(a,b)}$	$prefinf(a,b,[P])$
5	$\dfrac{has(a,[Q],n) \quad has(b,[Q'],m) \quad Q \approx Q' \succ P \quad n \neq m}{prefinf(a,b,[P]) \text{ is inapplicable}}$	$prefinf(a,b,[P])uc$
6	$\dfrac{has(a,[P],n) \quad has(b,[P'],m) \quad P \approx P' \quad n = m}{eqpref(a,b)}$	$eqprefinf(a,b,[P])$
7	$\dfrac{has(a,[Q],n) \quad has(b,[Q'],m) \quad Q \approx Q' \not\approx P \quad n \neq m}{eqprefinf(a,b,[P]) \text{ is inapplicable}}$	$eqprefinf(a,b,[P])uc$

Table 2. Inference schemes

3.2 Inferences

An argument is a derivation of a conclusion from a set of premises. Such a derivation is built from multiple steps called inferences. Every inference step consists of premises and a conclusion. Inferences can be chained by using the conclusion of one inference step as a premise in the following step. Thus a tree of chained inferences is created, which we use as the formal definition of an argument.

Definition 3. (Argument) *An* argument *is a tree, where the nodes are inferences, and an inference can be connected to a parent node if its conclusion is a premise of that node. Leaf nodes only have a conclusion (a formula from the knowledge base), and no premises. A subtree of an argument is also called a* subargument*. We define* `inf` *to be a function that returns the last inference of an argument (the root node), and* `conc` *to be a function that returns the conclusion of an argument, which is the same as the conclusion of the last inference.*

The inferences that can be made are defined by inference schemes. The inference schemes of our framework are listed in Table 2. The first and second inference schemes are used to count the number of attributes of equal importance as some attribute P that object a has. This type of inference is inspired by *accrual* [14], which combines multiple arguments with the same conclusion into one accrued argument for the same conclusion. Although our application is different, we use a similar mechanism. We want all attributes that are present to be counted. Otherwise we would conclude incorrect preferences (e.g. if the *large* attribute of the *apartment* were not counted, we would incorrectly derive that the *villa* were preferred over the *apartment*). Inference scheme 1, which counts 0, can always be applied since it has no premises. Inference scheme

$$A:\ \dfrac{\dfrac{large(apartment)\quad closeToWork(apartment)\quad large \approx closeToWork}{has(apartment,[large],2)}\quad \dfrac{garden(cottage)}{has(cottage,[garden],1)}\quad large \approx garden\quad 2>1}{pref(apartment,cottage)}$$

$$B:\ \dfrac{\dfrac{nearShops(apartment)}{has(apartment,[nearShops],1)}\quad \dfrac{}{has(villa,[nearShops],0)}\quad nearShops \approx nearShops\quad 1>0}{pref(apartment,villa)}$$

$$C:\ \dfrac{\dfrac{}{has(villa,[nearShops],0)}\quad \dfrac{}{has(apartment,[nearShops],0)}{}^{*}\quad nearShops \approx nearShops\quad 0=0}{eqpref(villa,apartment)}$$

$$D:\ \dfrac{nearShops(apartment)}{*\ \text{is inapplicable}}$$

Table 3. Example arguments.

2 can be applied on any subset of the set of attributes of some importance level on that an object a has. This means that it is possible to construct an argument that does not count all attributes that are present (a so-called non-maximal count). To ensure that only maximal counts are used, we provide an inference scheme to make arguments that defeat non-maximal counts (inference scheme 3). An argument of this type says that any count which is not maximal is not applicable. This type of defeat is called undercut (see below). Inference scheme 4 says that an object a is preferred over an object b if the number of attributes of a certain importance level that a has is higher than the number of attributes on that same level that b has. For the lexicographic ordering, it is also required that a and b have the same number of attributes on any level higher than that of P. We model this by defining an inference scheme 5 that undercuts scheme 4 if there is a more important level than that of P on which a and b do not have the same number of attributes. Finally, inference schemes 6 and 7 do the same as 4 and 5, but for equal preference. We need these because equal preference cannot be expressed in terms of strict preference.

Example 3. We now illustrate the inference schemes with some arguments that can be made from the knowledge base in Example 2. The example arguments are listed in Table 3 (for space reasons, the inference labels are left out). Argument A illustrates the general working; a preference for the apartment over the cottage is derived, based on the facts that the apartment has two attributes of some level and the cottage only one. Argument B illustrates a zero count. Here a preference for the apartment over the villa is derived, based on the facts that the apartment has one attribute of some level and the villa zero. In argument C a non-maximal count is used (stating that the apartment has zero attributes of the level of *nearShops*), which leads to another conclusion, namely that the villa and the apartment are equally preferred. However, there are undercutters to attack such arguments (argument D).

Note that the lexicographic ordering results in a complete transitive order of weak preference on objects. This means that it is not necessary to define inference rules for the property of transitivity, because any preference that follows from transitivity can also be derived directly from the definition of lexicographic ordering. For example, if $pref(a,b)$ and $eqpref(b,c)$ hold, then $pref(a,c)$ also holds, but this can be derived using the inference schemes of Table 2. The same holds for the asymmetry of strict preference and the symmetry of equal preference.

3.3 Defeat

With the language and the inference rules defined in the previous sections we can construct arguments. To complete our argumentation framework, we also need to specify a defeat relation. This section provides the formal definition of defeat that we will use. The most common type of defeat is rebuttal. An argument rebuts another argument if its conclusion is the negation of the conclusion of the other argument. Rebuttal is always mutual. Another type of defeat is undercut. An undercutter is an argument for the inapplicability of an inference used in another argument (for the specific undercutters used in our framework, see the next section). Undercut works only one way. Defeat is defined recursively, which means that rebuttal can attack an argument on all its premises and (intermediate) conclusions, and undercut can attack ist on all its inferences.

Definition 4. (Defeat) *An argument A* defeats *an argument B if*

- $\texttt{conc}(A) = \varphi$ *and* $\texttt{conc}(B) = \neg\varphi$ *(*rebuttal*), or*
- $\texttt{conc}(A) =$*'*$\texttt{inf}(B)$ *is inapplicable' (*undercut*), or*
- *A defeats a subargument of B.*

3.4 Semantics

By specifying the inference schemes and the definition of defeat, together with a knowledge base, we have instantiated an argumentation framework consisting of a set of arguments and a defeat relation among them. Now we define which arguments are justified. For this we use Dung's [10] grounded semantics.[1] Grounded semantics is defined as follows.

Definition 5.
- *An argument A is* acceptable *with respect to a set S of arguments iff each argument defeating A is defeated by an argument in S.*
- *The* characteristic function*, denoted by F_{AF}, of an argumentation framework AF is defined as follows:* $F_{AF}(S) = \{A \mid A \text{ is acceptable with respect to } S\}$.
- *The* grounded extension *of AF is defined as the least fixed point of F_{AF}.*
- *An argument is* justified *with respect to grounded semantics iff it is a member of the grounded extension.*

3.5 Validity

The argumentation framework defined in previous sections indeed models lexicographic preference, assuming a complete and consistent knowledge base.

Lemma 1. *Let $\mathscr{A}(KB)$ denote all arguments that can be built from a knowledge base KB. Then there is an argument $A \in \mathscr{A}(KB)$ such that the conclusion of A is pref(a,b) and A is justified under grounded semantics iff a is preferred over b according to the lexicographic preference ordering (Definition 1) given KB.*

[1] For the argumentation system defined in this paper (including the extended version of Section 5), the choice of semantics is not relevant; we could also have used other semantics such as preferred or stable semantics (also from [10]). There would be a difference when we allow the use of an inconsistent knowledge base, in which case another semantics may be more suitable. This is something for further investigation.

Proof. Suppose a is preferred over b. This means that there exists an attribute P such that $|\{P' \mid a \text{ satisfies } P' \text{ and } P \approx P'\}| > |\{P' \mid b \text{ satisfies } P' \text{ and } P \approx P'\}|$ and for all $Q \succ P$: $|\{Q' \mid a \text{ satisfies } Q' \text{ and } Q \approx Q'\}| = |\{Q' \mid b \text{ satisfies } Q' \text{ and } Q \approx Q'\}|$. Let $P_1 \dots P_n$ denote all attributes of equal importance as P such that a has P_i and let $P'_1 \dots P'_m$ denote all attributes of equal importance as P such that b has P_i. Note that $n > m$. Then the knowledge base is as follows: $P_1 \approx \dots \approx P_n \approx P'_1 \approx \dots P'_m$ and $P_1(a) \dots P_n(a)$ and $P'(b) \dots P'_m(b)$. The following argument (A) can be built (note that this argument can also be built if m is equal to 0, by using the empty set count):

$$\cfrac{\cfrac{P_1(a) \quad \dots \quad P_n(a) \quad P_1 \approx \dots \approx P_n}{\mathit{has}(a,[P_1],n)} \qquad \cfrac{P'_1(b) \quad \dots \quad P'_m(b) \quad P'_1 \approx \dots \approx P'_m}{\mathit{has}(b,[P'_1],m)} \qquad P_1 \approx P'_1 \quad n > m}{\mathit{pref}(a,b)}$$

We will now play devil's advocate and try to defeat this argument. We can try rebuttal and undercut of the argument and its subarguments. Rebuttal of premises is not applicable, since the knowledge base is consistent. Rebuttal of (intermediate) conclusions is not possible either, since there is no way to derive a negation. Then there are three inferences we can try to undercut (the last inference of the argument and the last inferences of two subarguments). For the left-hand count, this can only be done if there is another P_j such that $P_j \approx P$ and $P_j \notin \{P_1, \dots, P_n\}$ and $P_j(a)$ is the case. However, $P_1 \dots P_n$ encompass all such attributes, so count undercut is not possible. The same argument holds for the other count. At this point it is useful to note that these two counts are the only ones that are undefeated. Any lesser count will be undercut by the count undercutter that takes all of $P_1 \dots P_n$ (resp. $P'_1 \dots P'_m$) into account. Such an undercutter has no defeaters, so any non-maximal count is not justified. The final thing that is left to try is undercut of $\mathit{prefinf}(a,b,[P_1])$. The undercutter of $\mathit{prefinf}(a,b,[P_1])$ is based on two counts. We have seen that any non-maximal count will be undercut. If the maximal counts are used, we have $n = m$, since we have for all $Q \succ P$: $|\{Q' \mid a \text{ satisfies } Q' \text{ and } Q \approx Q'\}| = |\{Q' \mid b \text{ satisfies } Q' \text{ and } Q \approx Q'\}|$. So the undercutter inference rule cannot be applied since $n \neq m$ is not true. This means that for every possible type of defeat, either the defeat is inapplicable or the defeater of A is itself defeated by undefeated arguments. This means that A is in the grounded extension and hence justified according to grounded semantics. The same line of argument can be followed for $eqpref$. □

4 Strategies for Handling Incomplete Information

So far, we have defined an argumentation system that can reason about preferences according to the lexicographic preference ordering. Above, we have assumed that the information about the objects that are compared is complete. But, as stated in the introduction, this is often not the case. In this section we will investigate how incomplete information can best be handled when reasoning about preferences.

Suppose it is not known whether an object has a specific attribute, e.g. we know that $P(a)$ but we do not know whether $P(b)$ or $\neg P(b)$. This might not be a problem. If the preference between a and b can be decided based upon attributes that are more important than P, the knowledge whether $P(b)$ or $\neg P(b)$ is the case is irrelevant. But often this information will be needed to decide a lexicographic preference. In that case, different approaches or strategies for drawing conclusions are possible. However, not all strategies give desired results. In the following, we will discuss some naive strategies and their shortcomings, from which we will derive some desired properties of strategies, and define and model a strategy that gives intuitive results.

4.1 Naive Strategies

Optimistic resp. Pessimistic Strategy This strategy always assumes that an object has resp. does not have the attribute that is not known. This strategy can always derive some preference between two objects, since it completes the knowledge by making certain assumptions, and can then derive a complete preference ordering over objects. But there is no guarantee that the inferences made are correct. In fact, any inferred preference can only be correct if all the assumptions it is based on are either correct or irrelevant. Since we do not know whether assumptions are correct and the strategy does not check for relevance, the inference can only be correct by chance. For example, suppose it is not known whether the *villa* has a *garden* and whether it is *closeToWork*. The optimistic strategy would assume that it has both attributes, in which case an incorrect preference of the *villa* over the *apartment* would be derived. The pessimistic strategy on the other hand would assume the *villa* has neither of the attributes, and would derive an incorrect preference of the *cottage* over the *villa*.

Note that using the framework defined above without adaptation would boil down to using a pessimistic strategy: if it is not known whether an object as a certain attribute, the attribute is (implicitly) assumed to be absent. This is due to the fact that only attributes for which it is known that an object has them are counted. Attributes that an object does not have and attributes for which this information is unavailable are treated the same way (i.e. not taken into account when counting).

Disregard Attribute Strategy This strategy does not take into account the attributes for which information about the objects to be compared is incomplete. This strategy can always derive some preference between two objects, since the information regarding the remaining attributes is complete, so a complete preference ordering over objects can be derived. But the inference might not be correct, since the attributes that are disregarded might be relevant in defining a preference order. For example, suppose it is not known whether the *cottage* is *large*. In that case, the attribute *large* will not be taken into account when comparing the *cottage* to another object. This leaves only the attributes *garden* and *closeToWork* on the highest importance level, of which all attributes satisfy exactly one. Since the *cottage* has the most attributes on the next importance level, a preference of the *cottage* over the *villa* as well as the *apartment* will be derived, even though in the original example the *cottage* was the least preferred object.

Cautious Strategy In order to prevent the derivation of preferences that are only correct by chance, a natural alternative is to use a cautious strategy that prevents such inferences. This strategy infers nothing unless all information about the objects under comparison is available. It never makes incorrect preference inferences, but it lacks in decisiveness. Even if the unknown information is irrelevant to make an inference, nothing is inferred.

	P	Q	R
a	✓	✓	?
b	?		✓

a.

	P	Q
a	✓	?
b	?	✓

b.

	P	Q
a	✓	?
b		✓

c.

Table 4. Examples of objects and attributes with incomplete information

4.2 Desired Properties for Strategies

Given the limitations of the strategies discussed above, it is clear that we need a more balanced strategy that takes two main concerns into account, which we call decisiveness and safety.

Decisiveness We call a strategy *decisive* if it does not infer too little. As mentioned above, an unknown attribute might be irrelevant for deciding a preference. This is the case if the preference is already determined by more important attributes. For example, suppose that we do not know whether the *apartment* has attribute *nearShops*. Then we can still conclude that the *apartment* is preferred over the *cottage*, based on the attributes *large*, *garden*, and *closeToWork*. It is not required that a preference is derived in every case, since the missing information might be essential, but all preferences that are certain (for which no essential information is missing) should be derived. The cautious strategy is not decisive.

Safety We call a strategy *safe* if it does not infer too much. Suppose again that we do not know whether the *apartment* has attribute *nearShops*. Whereas this is irrelevant for deciding a preference between *apartment* and *cottage*, we do need this information for deciding the preference between the *villa* and the *apartment*. A strategy that makes assumptions about the missing information, or that disregards the attribute in question, will make unfounded inferences, and hence be unsafe. The optimistic, pessimistic and disregard attribute strategies are not safe.

4.3 A Decisive and Safe Strategy

We have seen above what may go wrong when a naive strategy is used to deal with incomplete information. In this section we define an alternative strategy that does satisfy the properties of decisiveness and safety identified above. A preference inference should never be based on an unfounded assumption for a strategy to be safe. But to be decisive, a strategy needs to be able to distinguish relevant from irrelevant information. Our approach is based on the following intuition. When comparing two objects under incomplete information, multiple situations are possible. That is, whenever it is not known whether an object has an attribute, there is a possibility that it does and a possibility that it does not. If a preference can be inferred in every possible situation, then apparently the missing information is not relevant, and it is safe to infer that preference. It is not necessary to check every possible situation, but it suffices to look at extreme cases. For every object, we can construct a best- and worst-case scenario, or best and worst possible situation. A possible situation is a *completion* of an object in the sense that all missing information is filled in.

Definition 6. (Completion) *A* completion *of an object a is an extension of the knowledge base with (previously missing) facts about a such that for every attribute P, either $P(a)$ or $\neg P(a)$ is in the extended knowledge base. So if a has n unspecified attributes, there are 2^n possible completions of a.*

Since we assumed that presence of an attribute is preferred over absence, the most preferred completion assumes presence of all unknown attributes, and the least preferred completion assumes absence. If even the least preferred completion of *a* is preferred over the most preferred completion of *b*, then *a* must always be preferred over *b*, since *a* could not be worse and *b* could not be better. For example, consider the objects and attributes in Table 4a. In the worst case for *a*, *a* does not have attribute *R*. In the best case for *b*, *b* has attribute *P*. But even in this situation, *a* will be preferred over *b*, based on attribute *Q*. There is no way that this situation can improve for *b* or deteriorate for *a*, so it is safe to infer a preference for *a* over *b*. The strategy's power to make such inferences makes it decisive.

The next example illustrates that this approach does not infer a preference when the missing information is relevant. Consider Table 4b. In the situation that is worst for *a* and best for *b*, *b* will be preferred because it has both attributes, while *a* only has *P*. But in the other extreme situation, that is best for *a* and worst for *b*, *a* is preferred. This means that in reality, anything is possible, and it is not safe to infer a preference.

We have seen when a preference for *a* over *b* can be inferred, and in which case no preference can be inferred. There are, however, two more possibilities. One is the case in which a preference of the most preferred completion of *a* over the least preferred completion of *b* can be derived, but only equal preference between the least preferred completion of *a* and the most preferred completion of *b*. This is illustrated in Table 4c. In this case, we would like to derive at least a weak preference of *a* over *b*. This is important, because in many cases a weak preference is strong enough to base a decision on, even if a strict preference cannot be derived. When having to decide between *a* and *b*, choosing *a* cannot be wrong when *a* is weakly preferred over *b*. Failing to derive a weak preference makes a strategy less decisive.

The last possibility is equal preference. We only want to derive an equal preference between two objects *a* and *b* if all possible completions of *a* are equally preferred as all possible completions of *b*. This also means that the most and least preferred completions of *a* and *b* have to be equally preferred. This can only be the case if all information about *a* and *b* is known, for as soon as some information is missing, there will be multiple possible completions which are not equally preferred.

5 Argumentation Framework for Incomplete Information

This section presents how our framework is extended to incorporate the decisive and safe strategy for incomplete information as presented in Section 4.3. We first present the changes to the language and then the changes to the inference rules. The defeat definition does not have to change.

5.1 Language

To distinguish between the different completions of an object, we introduce a completion label. We use the object name without label to denote the object in general, that is, the object with any completion. The superscript $^+$ is used for the most preferred completion of an object, $^-$ for the least preferred completion. For example, consider object a in Table 4a. The most preferred completion of a has attribute R, and is denoted a^+. The least preferred completion of a does not have attribute R, and is denoted a^-.

Reasoning with completions as discussed above can be viewed as a kind of assumption-based reasoning. To be able to support such reasoning, we extend the language and introduce weak negation, denoted by $\sim$, which is also used in [15]. This is used to formalize a kind of assumption-based reasoning. A formula $\sim \varphi$ can always be assumed, but is defeated by φ (see the next section for the details). So the statement $\sim \varphi$ should be interpreted as 'φ cannot be derived'.

Finally, we add formulas of the type $\mathit{wpref}(a,b)$ which express weak preference, just as $\mathit{pref}(a,b)$ and $\mathit{eqpref}(a,b)$ express strict and equal preference, respectively. We use weak preference in the sense that an object a is weakly preferred over an object b if any completion of a is either preferred over or equally preferred as any completion of b, but no strict or equal preference can be derived with certainty.

This leads to the following redefinition of the language.

Definition 7. **(Language)** *Let $\mathscr{P}$ be a set of attribute names with typical elements P, Q, and $\mathscr{O}$ a set of object names with typical elements a, b, and let n be a non-negative integer, and $x, y \in \{+, -, \{\}\}$ a label for objects (where $\{\}$ means no label). The* language *$\mathscr{L}$ is defined as follows.*

$$\varphi \in \mathscr{L} ::= P(a) \mid P \succ Q \mid P \approx Q \mid \mathit{pref}(a^x, b^y) \mid \mathit{eqpref}(a^x, b^y) \mid \mathit{wpref}(a^x, b^b) \mid$$
$$\mathit{has}(a^x, [P], n) \mid \neg\varphi \mid \sim \varphi$$

5.2 Inferences

The inference rules of the extended framework are listed in Table 5. Two inference rules are added that define the meaning of the weak negation $\sim$. According to inference rule 8, a formula $\sim \varphi$ can always be inferred, but such an argument will be defeated by an undercutter built with inference rule 9 if φ is the case.

P is supposed to be among the attributes of the least preferred completion of a (a^-) only if it is known that a has P. This is modelled by inference rule 2b in Table 5. For the most preferred completion of a, it is only required that it is not known that a does not have P; if this is not known, a^+ will be assumed to have P. This is modeled by using premises of the form $\sim \neg P(a)$ instead of $P(a)$. This can be seen in inference rule 2a. Inference rules 4 through 7 remain unchanged, except that completion labels are added.

To infer overall preferences from the preferences over certain completions, three more inference rules are defined. Inference rule 10 states that if (even) a^- is preferred over b^+, then a must be preferred over b, as we saw above. When a^+ is preferred over b^-, but a^- is only equally preferred as b^+, this not strong enough to infer a strict preference of a over b, but we can infer a weak preference of a over b using inference rule 11. Rule 12 states that in order to infer equal preference between a and b, both

1 $$\frac{}{has^x(a,[P],0)}\ count^x(a,[P],\varnothing)$$

2a $$\frac{\sim\neg P_1(a)\quad\ldots\quad\sim\neg P_n(a)\quad P_1\approx\ldots\approx P_n}{has(a^+,[P_1],n)}\ count(a^+,[P_1],\{P_1,\ldots P_n\})$$

2b $$\frac{P_1(a)\quad\ldots\quad P_n(a)\quad P_1\approx\ldots\approx P_n}{has(a^-,[P_1],n)}\ count(a^-,[P_1],\{P_1,\ldots P_n\})$$

3a $$\frac{\sim\neg P_1(a)\quad\ldots\quad\sim\neg P_n(a)\quad P_1\approx\ldots\approx P_n}{count(a^+,[P_1],S\subset\{P_1,\ldots,P_n\})\text{ is inapplicable}}\ count(a^+,[P_1],\{P_1,\ldots,P_n\})uc$$

3b $$\frac{P_1(a)\quad\ldots\quad P_n(a)\quad P_1\approx\ldots\approx P_n}{count(a^-,[P_1],S\subset\{P_1,\ldots,P_n\})\text{ is inapplicable}}\ count(a^-,[P_1],\{P_1,\ldots,P_n\})uc$$

4 $$\frac{has(a^x,[P],n)\quad has(b^y,[P'],m)\quad P\approx P'\quad n>m}{pref(a^x,b^y)}\ prefinf(a^x,b^y,[P])$$

5 $$\frac{has(a^x,[Q],n)\quad has(b^y,[Q'],m)\quad Q\approx Q'\succ P\quad n\neq m}{prefinf(a^x,b^y,[P])\text{ is inapplicable}}\ prefinf(a^x,b^y,[P])uc$$

6 $$\frac{has(a^x,[P],n)\quad has(b^y,[P'],m)\quad P\approx P'\quad n=m}{eqpref(a^x,b^y)}\ eqprefinf(a^x,b^y,[P])$$

7 $$\frac{has(a^x,[Q],n)\quad has(b^y,[Q'],m)\quad Q\approx Q'\not\approx P\quad n\neq m}{eqprefinf(a^x,b^y,[P])\text{ is inapplicable}}\ eqprefinf(a^x,b^y,[P])uc$$

8 $$\frac{}{\sim\varphi}\ asm(\sim\varphi)$$

9 $$\frac{\varphi}{asm(\sim\varphi)\text{ is inapplicable}}\ asm(\sim\varphi)uc$$

10 $$\frac{pref(a^-,b^+)}{pref(a,b)}$$

11 $$\frac{eqpref(a^-,b^+)\quad pref(a^+,b^-)}{wpref(a,b)}$$

12 $$\frac{eqpref(a^+,b^-)\quad eqpref(a^-,b^+)}{eqpref(a,b)}$$

Table 5. Inference schemes for incomplete information

the most preferred completion of a and the least preferred completion of b, and the least preferred completion of a and the most preferred completion of b must be equally preferred.

Example 4. In the case of Table 4a, the following argument can be built.

$$\dfrac{\dfrac{\dfrac{\dfrac{Q(a)}{has(a^-,[Q],1)}\quad \dfrac{}{has(b^+,[Q],0)}\quad Q\approx Q\quad 1>0}{pref(a^-,b^+)}}{pref(a,b)}}{}$$

The next argument shows that a weak preference can be inferred in the situation of Table 4c.

$$\dfrac{\dfrac{\dfrac{P(a)}{has(a^-,[P],1)} \quad \dfrac{\overline{\sim\neg Q(b)}}{has(b^+,[Q],1)} \quad P\approx Q \quad 1=1}{eqpref(a^-,b^+)} \quad \dfrac{\dfrac{\overline{\sim\neg P(a)} \quad \overline{\sim\neg Q(a)} \quad P\approx Q}{has(a^+,[P],2)} \quad \dfrac{Q(b)}{has(b^-,[Q],1)} \quad P\approx Q \quad 2>1}{pref(a^+,b^-)}}{wpref(a,b)}$$

6 Conclusion

In this paper we have made the following contributions. Argumentation-based approaches can be used to model qualitative multi-attribute preferences such as the lexicographic ordering. The advantage of argumentation over other approaches emerges most clearly in the case of incomplete information. Our approach allows to reason about preferences from best- and worst-case perspectives (called completions here), and the consequences for overall preferences.

In our current approach it is still often the case that no preference can be inferred. What should we do in such a case? One approach is to ask the user for the missing information. But the user might not have this information, and might not have the time or resources to look it up. In some situations it might be fruitful to relax the notion of safety, which we have used in a very strict sense here; a conclusion is only called safe if it can be drawn in every possible situation. But we might want to draw a conclusion if it follows in the most likely situation. Of course, to model this we need information about the likelihood of situations. This could for example be modelled by a normality ranking [3] or a possibility ranking [9]. Also, although general default assumptions are often not safe, some domain-specific default assumptions may be safe enough. For example, if nothing to the contrary is known, one may safely assume that a house has electricity. Some default assumptions may be conditional, for example, a detached house usually has a garden. One interesting extension therefore is to add such default reasoning and more general reasoning about the beliefs of an agent to the framework. Default rules (e.g. $detached(a) \Rightarrow garden(a)$) can be placed in the knowledge base. Next, an inference rule is needed that applies these rules and can infer $garden(a)$ from $detached(a)$ and $detached(a) \Rightarrow garden(a)$. Finally, a strength mechanism is needed, so that factual information always defeats rebutting default assumptions (e.g. if $\neg garden(a)$ is known for a fact, then this defeats the conclusion $garden(a)$ that was derived using a default rule, but not vice versa).

In our future work we would like to distinguish more explicitly between mental attitudes such as beliefs, goals, desires and preferences. This will also allow us to reason about these attitudes, for example that a certain preference we have is based on some specific beliefs. We hope to gain insight from modal preference languages with belief operators such as the one presented in [13]. Other interesting areas for future work include the representation of dependent preferences (e.g. 'I only want a balcony if the house does not have a garden, otherwise I do not care'), and the relation with e.g. CP-nets [4] and value-based argumentation [11].

Finally, we believe that the argumentation-based framework for preferences presented here can be usefully applied in the preference elicitation process. It allows the user to extend and refine the system representation of his preferences gradually and as the user sees fit. To facilitate this elicitation process more research is needed how our framework can support a user e.g. by indicating which information is still missing.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075.

References

1. L. Amgoud, N. Maudet and S. Parsons. Modelling dialogues using argumentation. *Proc. ICMAS*, 2000.
2. T.J.M. Bench-Capon and P.E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.
3. C. Boutilier. Toward a logic for qualitative decision theory. *Proc. KR*, pages 75–86, 1994.
4. C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
5. G. Brewka. A rank based description language for qualitative preferences. *Proc. ECAI*, 2004.
6. G. Brewka, S. Benferhat, and D. Le Berre. Qualitative choice logic. *Artificial Intelligence*, 157(1-2):203–237, 2004.
7. S. Coste-Marquis, J. Lang, P. Liberatore, and P. Marquis. Expressive power and succinctness of propositional languages for preference representation. *Proc. KR*, pages 203–212, 2004.
8. J. Doyle and R.H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68, 1999.
9. D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. *Proc. IJCAI*, 1995.
10. P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence*, 77:321–357, 1995.
11. S. Kaci and L. van der Torre. Preference-based argumentation: Arguments supporting multiple values. *Int. J. of Approximate Reasoning*, 48:730–751, 2008.
12. R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. Cambridge University Press, 1993.
13. F. Liu. *Changing for the Better: Preference Dynamics and Agent Diversity*. PhD thesis, Universiteit van Amsterdam, 2008.
14. H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. *Proc. ICAIL*, pages 85–94, 2005.
15. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
16. I. Rahwan, S.D. Ramchurn, N.R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4):343–375, 2004.