An Argumentation Framework for Qualitative Multi-Criteria Preferences

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

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

Abstract. Preferences between different alternatives (products, decisions, agreements etc.) are often based on multiple criteria. Qualitative Preference Systems (QPS) is a formal framework for the representation of qualitative multi-criteria preferences in which a criterion's preference is defined based on the values of attributes or by combining multiple subcriteria in a cardinality-based or lexicographic way. In this paper we present a language and reasoning mechanism to represent and reason about such qualitative multi-criteria preferences. We take an argumentation-based approach and show that the presented argumentation framework correctly models a QPS. Then we extend this argumentation framework in such a way that it can derive missing information from background knowledge, which makes it more flexible in case of incomplete specifications.

1 Introduction

In the context of practical reasoning, such as decision making and negotiation, preferences between the available alternatives play a key role. A system supporting a human user in such tasks should therefore have a representation of that user's preferences. In this paper we present an argumentation framework to represent and reason with *qualitative*, *multi-criteria* preferences. Preferences are modelled in a qualitative way because it is hard for humans to give exact numeric utilities. We use multiple criteria because it is a very natural thing to compare two alternatives on several criteria and base an overall preference on those comparisons. Criteria thus represent the *underlying interests*, or *reasons* for preferences. Moreover, the outcome space may be so large that it is infeasible to specify preference between outcomes directly.

We briefly present a framework for representing qualitative multi-criteria preferences, called Qualitative Preference Systems. In this framework, preferences between outcomes are determined by combining multiple criteria based on cardinality and lexicographic ordering. Ultimately, the criteria are based on preferences between the values of relevant variables. QPS is a framework that provides a formal definition of qualitative multi-criteria preferences. The aim of this paper is to provide a *language* and *reasoning mechanism* to reason about such qualitative preference systems. In addition, we provide the means of deriving information by default from background knowledge, which is useful when e.g. the outcomes are incompletely specified.

The approach we take is argumentation-based. Argumentation is a kind of defeasible reasoning, which allows for reasoning with incomplete information in a commonsense way, about things that are normally the case. Moreover, argumentation is a natural way of reasoning for humans. As such, it is suitable for explaining the reasoning of a system to a human user. Finally, argumentation can be used in a persuasion dialogue, for example when multiple agents with different preferences have to agree on a common action.

Note that the argumentation framework presented here is *not* a preference-based argumentation framework (PAF) ins the sense of [1]. In a PAF, preference between arguments are used to determine the success of an attack between them. A similar approach, that considers preferences between rules in the logical language, has been taken in the specific context of decision making [2]. In contrast, the framework presented here aims to reason about preferences between objects outside of the argumentation framework ('outcomes') as opposed to preferences between arguments or logical rules.

The outline of the paper is as follows. In Section 2, we briefly recall qualitative preference systems. Section 3 presents the argumentation framework that provides the means to reason about a QPS. In Section 4 we extend the argumentation framework with background knowledge and the means to derive information by default. Finally, Section 5 concludes the paper.

2 Qualitative Preference Systems

In this section we briefly present qualitative preference systems. The main aim of a QPS is to determine preferences between *outcomes* (or *alternatives*). An outcome is represented as an assignment of values to a set of relevant variables. Every variable has its own domain of possible values. Constraints on the assignments of values to variables are expressed in a knowledge base. Outcomes are defined as variable assignments that respect the constraints in the knowledge base.

The preferences between outcomes are based on multiple *criteria*. Every criterion can be seen as a *reason* for preference, or as a preference from one particular *perspective*. A distinction is made between simple and compound criteria. Simple criteria are based on a single variable. Multiple (simple) criteria can be combined in order to determine an overall preference. In a QPS, this is done with compound criteria. There are two kinds of compound criteria: cardinality criteria and lexicographic criteria. The subcriteria of a cardinality criterion all have equal importance, and preference is determined by counting the number of subcriteria that support it. In a lexicographic criterion, the subcriteria are ordered by priority and preference is determined by the most important subcriteria.

Definition 1. (Qualitative preference system) A *qualitative preference system (QPS)* is a tuple $\langle Var, Dom, K, \Omega, \mathcal{C} \rangle$. *Var* is a finite set of *variables*. Every variable $X \in Var$ has a domain $Dom(X)$ of possible values. K is a set of constraints on the assignments of values to the variables in *Var*. Ω is the set of all outcomes. An *outcome* α is an assignment of a value $x \in Dom(X)$ to every variable $X \in Var$, such that no constraints in K are violated. α_X denotes the value of variable X in outcome α. $\mathcal{C} = \mathcal{C}_s \cup \mathcal{C}_c \cup \mathcal{C}_l$ is a set of criteria, where $\mathcal{C}_s$ contains simple criteria, $\mathcal{C}_c$ contains cardinality criteria and $\mathcal{C}_l$ contains lexicographic criteria. Weak preference between outcomes by a criterion c is denoted by the relation $\succeq_c$. $\succ_c$ denotes the strict subrelation, $\approx_c$ the indifference subrelation.

Definition 2. (Simple criterion) A *simple criterion* c is a tuple $\langle X_c, \geq_c \rangle$, where $X_c \in Var$ is a variable, and $\geq_c$, a preference relation on the possible values of X_c, is a preorder on $Dom(X_c)$. A simple criterion $c = \langle X_c, \geq_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \geq_c \beta$, if $\alpha_{X_c} \geq_c \beta_{X_c}$.

Definition 3. (Cardinality criterion) A *cardinality criterion* c is a tuple $\langle C_c \rangle$ where C_c is a nonempty set of criteria (the *subcriteria* of c). A cardinality criterion $c = \langle C_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \geq_c \beta$, iff $|\{s \in C_c \mid \alpha >_s \beta\}| \geq |\{s \in C_c \mid \alpha \ngeq_s \beta\}|$.

Definition 4. (Lexicographic criterion) A *lexicographic criterion* c is a tuple $\langle C_c, \rhd_c \rangle$, where C_c is a nonempty set of criteria (the *subcriteria* of c) and $\rhd_c$, a *priority relation* among subcriteria, is a strict partial order (a transitive and asymmetric relation) on C_c. A lexicographic criterion $c = \langle C_c, \trianglerighteq_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \geq_c \beta$, if $\forall s \in C_c (\alpha \geq_s \beta \vee \exists s' \in C_c (\alpha >_{s'} \beta \wedge s' \rhd_c s))$.

3 Argumentation Framework

In this section we present an argumentation framework for reasoning about qualitative multi-criteria preferences as defined in qualitative preference systems. The AF provides the logical language to represent facts about outcomes, criteria and preferences, and the means to construct arguments that infer preferences from certain input.

3.1 Abstract Argumentation Framework

Our argumentation framework is a concrete instantiation of an abstract argumentation framework as defined by Dung [3]. To define which arguments are justified, we use Dung's preferred semantics.

Definition 5. (Abstract argumentation framework) An *abstract argumentation framework (AF)* is a pair $\langle \mathcal{A}, \rightarrow \rangle$ where $\mathcal{A}$ is a set of *arguments* and $\rightarrow$ is a *defeat* relation among those arguments.

Definition 6. (Preferred semantics) A *preferred extension* of an AF $\langle \mathcal{A}, \rightarrow \rangle$ is a maximal (w.r.t. $\subseteq$) set $S \subseteq \mathcal{A}$ such that: $\forall A, B \in S : A \not\rightarrow B$ and $\forall A \in S$: if $\exists B \in \mathcal{A} : B \rightarrow A$ then $\exists C \in S : C \rightarrow B$. An argument is credulously (resp. sceptically) *justified* w.r.t. preferred semantics if it is in some (resp. all) preferred extension(s). An argument is overruled if it is not in any extension. We also say that a formula is justified (resp. overruled) iff it is the conclusion of a justified (resp. overruled) argument.

An abstract AF can be instantiated by specifying the *structure of arguments* and the *nature of the defeat relation*. Prakken [4] presents such an instantiation that is itself still abstract: his *argumentation systems* define arguments as inference trees formed by applying inference rules and specify three kinds of defeat. We take the instantiation of an argumentation framework one step further and also define the *logical language* and the specific *inference schemes* that are used.

3.2 Arguments

Arguments are built from formulas of a logical language, that are chained together using inference steps. 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 (cf. e.g. [5, 4]).

Definition 7. (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*. inf returns the last inference of an argument (the root node), and conc returns the conclusion of an argument, which is the same as the conclusion of the last inference.

3.3 Defeat

We define two different kinds of defeat: rebuttal and undercut (note that, unlike e.g. [4], in the current framework there is no distinction between *attack* and *defeat*). An argument *rebuts* another argument if its conclusion contradicts a conclusion of the other argument. Which conclusions contradict each other is defined below after the language is introduced. Defeat by rebuttal is mutual. The term undercut is used in different ways in the literature; we use it for the same concept as e.g. [4]. An *undercutter* is an argument for the inapplicability of an inference step made in another argument. Hence, it is a kind of meta-reasoning (the conlusion of an undercutting argument is not part of the object language). 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 it on all its inferences.

Definition 8. (Defeat) An argument A *defeats* an argument B ($A \rightarrow B$) if $\texttt{conc}(A)$ and $\texttt{conc}(B)$ are contradictory (*rebuttal*), or $\texttt{conc}(A) =$'$\texttt{inf}(B)$ is inapplicable' (*undercut*), or A defeats a subargument of B.

3.4 Language

The logical language provides the means to express statements about a the elements of a QPS. For a given QPS $S = \langle Var, Dom, K, \Omega, \mathcal{C} \rangle$, the *domain of discourse* is $D = Var \cup \bigcup_{X \in Var} Dom(X) \cup \Omega \cup \mathcal{C}$, i.e. variables and their possible values, outcomes and criteria.

We make a distinction between an *input* and *full* language. A knowledge base, which is the input for an argumentation framework, is specified in the input language. The input language allows us to express facts about the outcomes that are considered and details about the criteria that are used. With the full language we can also express preferences. Such statements can be *derived* from a knowledge base with the inference rules that will be introduced in the next section.

Basic expressions of the language (*atoms*) are built from predicates and terms. Let C be a set of *constants*. $i : C \mapsto D$ is an *interpretation function* that assigns an element

predicate	interpretation	
$\mathtt{val}(o,x,y)$	$i(o)_{i(x)} = i(y)$ 'the value of variable x in outcome o is y'	where $i(o) \in \Omega$, $i(x) \in Var$, $i(y) \in Dom(i(x))$
$\mathtt{sc}(c,x)$	$i(c) \in \mathcal{C}_s$, $X_{i(c)} = i(x)$ where $i(x) \in Var$, 'c is a simple criterion on variable x'	
$\mathtt{valpref}(c,y_1,y_2)$	$i(y_1) \geq_{i(c)} i(y_2)$ 'simple criterion c weakly prefers value y_1 over value y_2'	where $i(c) \in \mathcal{C}_s$, $i(y_1), i(y_2) \in Dom(X_{i(c)})$
$\mathtt{cc}(c)$	$i(c) \in \mathcal{C}_c$ 'c is a cardinality criterion'	
$\mathtt{lc}(c)$	$i(c) \in \mathcal{C}_l$ 'c is a lexicographic criterion'	
$\mathtt{sub}(c,c_1)$	$i(c_1) \in C_{i(c)}$ 'c_1 is a subcriterion of criterion c'	where $i(c) \in \mathcal{C}_c \cup \mathcal{C}_l$, $i(c_1) \in \mathcal{C}$
$\mathtt{prior}(c,c_1,c_2)$	$i(c_1) \triangleright_{i(c)} i(c_2)$ 'subcriterion c_1 has higher priority than subcriterion c_2 according to lexicographic criterion c'	where $i(c) \in \mathcal{C}_l$, $i(c_1), i(c_2) \in \mathcal{C}$

Table 1. The predicates in $\mathcal{P}_{in}$ and their interpretation

from the domain of discourse to every constant in C. There are two sets of *predicates*. $\mathcal{P}_{in}$ contains predicates that can be used in the input language. $\mathcal{P}_{out}$ contains predicates that cannot be used in the input language and can only be derived. The predicates in $\mathcal{P}_{in}$ and $\mathcal{P}_{out}$ and their interpretation are in Table 1 and 2.

Formulas of the input language are just atoms of the input language. Formulas of the full language are atoms (A) or weakly negated atoms ($\sim A$). Weak negation is negation as failure: $\sim A$ is justified if A is not. Strong negation is not needed to model qualitative preference systems, but it will be added in the extended version of the AF presented in Section 4 in order to reason with background knowledge.

Definition 9. (Language) The input language is defined as follows.
atom$_{in}$ ::= $p(t_1, \ldots, t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{in}$
literal$_{in}$ ::= *atom*$_{in}$
formula$_{in}$ ::= *literal*$_{in}$

The full language is defined as follows.
atom$_{out}$ ::= $p(t_1, \ldots, t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{out}$
literal ::= *literal*$_{in}$ | *atom*$_{out}$
formula ::= *literal* | $\sim$ *literal*

Contradictory formulas Two arguments rebut each other if their conclusions are contradictory. There are two ways in which two formulas can be contradictory.

- The formulas specify different values for the same variable in the same outcome: $\mathtt{val}(o,x,y)$ and $\mathtt{val}(o,x,y')$ contradict each other if $y \neq y'$.
- $\mathtt{prior}(c,c_1,c_2)$ and $\mathtt{prior}(c,c_2,c_1)$ contradict each other, since priority is asymmetric.

predicate	interpretation
`pref(`c`,`o_1`,`o_2`)`	$i(o_1) \succeq_{i(c)} i(o_2)$ where $i(c) \in \mathcal{C}$, $i(o_1), i(o_2) \in \Omega$ 'criterion c weakly prefers outcome o_1 over outcome o_2'
`spref(`c`,`o_1`,`o_2`)`	$i(o_1) \succ_{i(c)} i(o_2)$ where $i(c) \in \mathcal{C}$, $i(o_1), i(o_2) \in \Omega$ 'criterion c strictly prefers outcome o_1 over outcome o_2'
`epref(`c`,`o_1`,`o_2`)`	$i(o_1) \approx_{i(c)} i(o_2)$ where $i(c) \in \mathcal{C}$, $i(o_1), i(o_2) \in \Omega$ 'criterion c equally prefers outcome o_1 and outcome o_2'
`sp(`c`,`o_1`,`o_2`,`n`)`	$\lvert\{s \in C_{i(c)} \mid i(o_1) \succ_s i(o_2)\}\rvert = n$ where $i(c) \in \mathcal{C}_c$, $i(o_1), i(o_2) \in \Omega$ 'there are n subcriteria of cardinality criterion c that strictly prefer outcome o_1 over outcome o_2'
`nwp(`c`,`o_1`,`o_2`,`n`)`	$\lvert\{s \in C_{i(c)} \mid i(o_1) \not\succeq_s i(o_2)\}\rvert = n$ where $i(c) \in \mathcal{C}_c$, $i(o_1), i(o_2) \in \Omega$ 'there are n subcriteria of cardinality criterion c that do not weakly prefer outcome o_1 over outcome o_2'

Table 2. The predicates in $\mathcal{P}_{out}$ and their interpretation

Two other candidates for contradiction are not modelled as such because they are handled in a different way.

One might argue that φ and $\sim \varphi$ are contradictory, and hence arguments concluding them should rebut each other. However, the status of these conclusions is not equal. φ has to be derived and is grounded in facts in the knowledge base. $\sim \varphi$ on the other hand is an assumption that can be made in the absence of evidence to the contrary. φ is such evidence to the contrary, and that is why an argument concluding φ undercuts the inference of $\sim \varphi$ instead of rebutting the conclusion (see the inference schemes for weak negation and its undercutter below).

Incompatible preference statements, such as e.g. `spref(c,o1,o2)` and `epref(c,o1,o2)` will resolve because `epref(c,o1,o2)` can only be derived if `pref(c,o2,o1)`, in which case the ~`pref(c,o2,o1)` premise needed to derive `spref(c,o1,o2)` will be undercut. Hence to have such arguments rebut each other would be superfluous.

Input knowledge base An input knowledge base is a set of formulas of the input language. A knowledge base *KB corresponds to* a QPS $S = \langle Var, Dom, K, \Omega, \mathcal{C} \rangle$ if the following condition holds: a formula φ is in *KB* iff its interpretation holds in *S*. Note that a knowledge base corresponding to a QPS is conflict-free, i.e. does not contain contradictory formulas.

Example 1. *We will use a running example throughout the paper to illustrate the details of the argumentation framework. Anne is planning to go on holiday with a friend. Anne's overall preference is based on three simple criteria:* `c1`*: that someone (she or the accompanying friend) speaks the language (*`sl`*),* `c2`*: that it is sunny (*`su`*) and* `c3`*: that she has not been there before (*`bb`*).* `c1` *and* `c2` *have equal priority, so they are aggregated in a cardinality criterion* `c4`*.* `c3` *and* `c4` *are combined in a lexicographic criterion* `c5` *where* `c3` *has higher priority than* `c4`*. This information can be represented in the following knowledge base.*

Facts about two of the possible outcomes:

```
val(o1,sl,true)     val(o1,su,true)     val(o1,bb,true)
val(o2,sl,false)    val(o2,su,true)     val(o2,bb,false)
```

Information about the preferences:

```
lc(c5)          cc(c4)          sc(c1,sl)      valpref(c1,true,false)
sub(c5,c3)      sub(c4,c1)      sc(c2,su)      valpref(c2,true,false)
sub(c5,c4)      sub(c4,c2)      sc(c3,bb)      valpref(c3,false,true)
prior(c5,c3,c4)
```

3.5 Inference rules

In this section we present the inference rules that are used in the argumentation framework to build arguments.

Weak negation The following two inference rules make sure that (i) a weakly negated formula can always be derived, but (ii) this inference will be undercut if the formula itself can be derived. So $\sim\varphi$ is sceptically justified iff φ is overruled.

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

Strict and equal preference The following inference schemes are used to derive strict and equal preference from weak preference according to the common definitions.

$$\frac{\mathtt{pref}(c,o_1,o_2)\quad \mathtt{\sim pref}(c,o_2,o_1)}{\mathtt{spref}(c,o_1,o_2)} \qquad\qquad \frac{\mathtt{pref}(c,o_1,o_2)\quad \mathtt{pref}(c,o_2,o_1)}{\mathtt{epref}(c,o_1,o_2)}$$

Preference by a simple criterion The following inference rule concludes that a simple criterion prefers one outcome over another if, for the variable that it is based on, it prefers the value of the first outcome over the value of the second. This is exactly the definition of preference by a simple criterion in a QPS.

$$\frac{\mathtt{sc}(c,x)\quad \mathtt{val}(o_1,x,y_1)\quad \mathtt{val}(o_1,x,y_2)\quad \mathtt{valpref}(c,y_1,y_2)}{\mathtt{pref}(c,o_1,o_2)}$$

Example 2. *The following argument infers that simple criterion* `c1` *prefers* `o1` *over* `o2`*. Similar arguments can be constructed for* `c2` *and* `c3`*.*

$$\frac{\mathtt{sc(c1,sl)}\quad \mathtt{val(o1,sl,true)}\quad \mathtt{val(o2,sl,false)}\quad \mathtt{valpref(c1,true,false)}}{\mathtt{pref(c1,o1,o2)}}$$

Preference by a cardinality criterion The next inference scheme derives preference by a cardinality criterion according to its definition in a QPS: an outcome o_1 is weakly preferred over an outcome o_2 if there are at least as many subcriteria that strictly prefer o_1 over o_2 as subcriteria that do not weakly prefer o_1 over o_2.

$$\frac{\mathtt{cc}(c)\quad \mathtt{sp}(c,o_1,o_2,l)\quad \mathtt{nwp}(c,o_1,o_2,m)\quad l\geq m}{\mathtt{pref}(c,o_1,o_2)}$$

Preference by a cardinality criterion is based on (i) the number of subcriteria that strictly prefer one outcome over the other, and (ii) the number of subcriteria that do not weakly prefer one outcome over the other. The following inference rules provide the required counting mechanism.

The next inference rules conclude that there are n subcriteria of c that strictly prefer o_1 over o_2, resp. that there are n subcriteria of c that do not weakly prefer o_1 over o_2.

$$\frac{\texttt{spref}(c_1,o_1,o_2) \quad \dots \quad \texttt{spref}(c_n,o_1,o_2) \quad \texttt{sub}(c,c_1) \quad \dots \quad \texttt{sub}(c,c_n)}{\texttt{sp}(c,o_1,o_2,n)}\ SP(c,o_1,o_2,n)$$

$$\frac{{\sim}\texttt{pref}(c_1,o_1,o_2) \quad \dots \quad {\sim}\texttt{pref}(c_n,o_1,o_2) \quad \texttt{sub}(c,c_1) \quad \dots \quad \texttt{sub}(c,c_n)}{\texttt{nwp}(c,o_1,o_2,n)}\ NWP(c,o_1,o_2,n)$$

If there are no subcriteria of c that strictly prefer o_1 over o_2, resp. that do not weakly prefer o_1 over o_2, no premises are needed to infer this.

$$\frac{}{\texttt{sp}(c,o_1,o_2,\texttt{0})}\ SP(c,o_1,o_2,\texttt{0}) \qquad\qquad \frac{}{\texttt{nwp}(c,o_1,o_2,\texttt{0})}\ NWP(c,o_1,o_2,\texttt{0})$$

With these inference schemes, it is possible to derive a formula $\texttt{sp}(c,o_1,o_2,n)$ for any n between 0 and the actual number of subcriteria of c that strictly prefer o_1 over o_2. We want to make sure that only the formula that counts *all* subcriteria of c that strictly prefer o_1 over o_2 is justified. To this end, the following inference rules provide an undercutter for the previous schemes when they are non-maximal.

$$\frac{\texttt{spref}(c_1,o_1,o_2) \quad \dots \quad \texttt{spref}(c_n,o_1,o_2) \quad \texttt{sub}(c,c_1) \quad \dots \quad \texttt{sub}(c,c_n) \quad m<n}{SP(c,o_1,o_2,m) \text{ is inapplicable}}\ SP(c,o_1,o_2,m)uc$$

$$\frac{{\sim}\texttt{pref}(c_1,o_1,o_2) \quad \dots \quad {\sim}\texttt{pref}(c_n,o_1,o_2) \quad \texttt{sub}(c,c_1) \quad \dots \quad \texttt{sub}(c,c_n) \quad m<n}{NWP(c,o_1,o_2,m) \text{ is inapplicable}}\ NWP(c,o_1,o_2,m)uc$$

Example 3. *The following argument concludes that there is one subcriterion of* `c4` *that strictly prefers* `o1` *over* `o2`*.*

$$\frac{\dfrac{\dfrac{\vdots}{\texttt{pref(c1,o1,o2)}} \quad \dfrac{}{{\sim}\texttt{pref(c1,o2,o1)}}}{\texttt{spref(c1,o1,o2)}} \qquad \texttt{sub(c4,c1)}}{\texttt{sp(c4,o1,o2,1)}}$$

It is also possible to construct an argument stating that there are two such criteria, but it will be undercut.

$$\frac{\dfrac{\dfrac{\vdots}{\texttt{pref(c1,o1,o2)}} \quad \dfrac{}{{\sim}\texttt{pref(c1,o2,o1)}}}{\texttt{spref(c1,o1,o2)}} \quad \dfrac{\dfrac{\vdots}{\texttt{pref(c2,o1,o2)}} \quad \dfrac{}{{\sim}\texttt{pref(c2,o2,o1)}}\ *}{\texttt{spref(c2,o1,o2)}} \qquad \texttt{sub(c4,c1)} \quad \texttt{sub(c4,c2)}}{\texttt{sp(c4,o1,o2,2)}}$$

$$\dfrac{\dfrac{\vdots}{\texttt{pref(c2,o2,o1)}}}{* \text{ is inapplicable}}$$

The following argument concludes that c4 *prefers* o1 *over* o2*.*

$$\dfrac{\dfrac{\vdots}{\texttt{sp(c4,o1,o2,1)}} \quad \dfrac{}{\texttt{nwp(c4,o1,o2,0)}} \quad 1 \geq 0}{\texttt{pref(c4,o1,o2)}}$$

Preference by a lexicographic criterion The following inference rule concludes that a lexicographic criterion c prefers an outcome o_1 over an outcome o_2 if o_1 is preferred over o_2 by a subcriterion of c. This inference is undercut by the next inference rule if there is a subcriterion of c with higher priority that does not prefer o_1 over o_2.

$$\dfrac{\texttt{lc(}c\texttt{)} \quad \texttt{sub(}c,c_1\texttt{)} \quad \texttt{pref(}c_1,o_1,o_2\texttt{)}}{\texttt{pref(}c,o_1,o_2\texttt{)}} \; LC(c,c_1,o_1,o_2)$$

$$\dfrac{\texttt{lc(}c\texttt{)} \quad \texttt{sub(}c,c_2\texttt{)} \quad \texttt{\~{}pref(}c_2,o_1,o_2\texttt{)} \quad \texttt{\~{}prior(}c,c_1,c_2\texttt{)}}{LC(c,c_1,o_1,o_2) \text{ is inapplicable}} \; LC(c,c_1,o_1,o_2)uc$$

According to its definition in a QPS, a lexicographic criterion c prefers o_1 over o_2 if every subcriterion either (weakly) prefers o_1 over o_2 or there is a higher priority subcriterion that strictly prefers o_1 over o_2. So if c prefers o_1 to o_2, all undominated (w.r.t. priority) subcriteria prefer o_1 to o_2. pref(c,o_1,o_2) can be derived based on any of those subcriteria, and there will be no justified undercutter. If c does not prefer o_1 to o_2, it may still be possible to construct an argument for pref(c,o_1,o_2), but it will be undercut because there is another subcriterion that does not prefer o_1 to o_2 and does not have lower priority. So together this pair of inference schemes correctly models the definition of preference by a lexicographic criterion in a QPS.

Example 4. *The following argument concludes that* c5 *prefers* o1 *to* o2 *based on its subcriterion* c4*.*

$$\dfrac{\texttt{lc(c5)} \quad \texttt{sub(c5,c4)} \quad \dfrac{\vdots}{\texttt{pref(c4,o1,o2)}}}{\texttt{pref(c5,o1,o2)}} \; *$$

However, this argument is undercut by the following one stating that there is another subcriterion, c3*, that does not prefer* o1 *to* o2 *and does not have lower priority than* c4*.*

$$\dfrac{\texttt{lc(c5)} \quad \texttt{sub(c5,c3)} \quad \dfrac{}{\texttt{\~{}pref(c3,o1,o2)}} \quad \dfrac{}{\texttt{\~{}sprior(c5,c4,c3)}}}{* \text{ is inapplicable}}$$

The only justified argument for preference between o1 *and* o2 *by* c5 *is the following one.*

$$\dfrac{\texttt{lc(c5)}\quad \texttt{sub(c5,c3)}\quad \dfrac{\texttt{sc(c3,bb)}\quad \texttt{val(o2,bb,false)}\quad \texttt{val(o1,bb,true)}\quad \texttt{valpref(c3,false,true)}}{\texttt{pref(c3,o2,o1)}}}{\texttt{pref(c5,o2,o1)}}$$

3.6 Correspondence between QPS and AF

Theorem 1. Let $S = \langle Var, Dom, K, \Omega, \mathcal{C} \rangle$ be a QPS, KB a knowledge base that corresponds to S, and AF the argumentation framework built from KB. Then φ is a sceptically justified conclusion of AF iff its interpretation holds in S.

For every formula in KB, its interpretation holds in S (definition of correspondence). Every formula in the input language whose interpretation holds in S is in KB (definition of correspondence). All formulas in KB are justified since KB is conflict-free. For every inference rule, its conclusion is justified if and only if its premises are justified and all its undercutters (if any) are overruled. We have shown that every inference or pair of inference and its undercutter inference models the corresponding QPS definition: the interpretation of the conclusion holds in a QPS if and only if the interpretations of all premises hold and and the interpretations of the premises of all undercutters do not all hold.

4 Reasoning with Background Knowledge

The argumentation framework presented in the previous section models a QPS if the input is a knowledge base corresponding to that QPS. In order for a knowledge base to correspond to a QPS, it is necessary to specify the values of all variables for every outcome. This correpsonds to the formal (abstract) concept of an outcome as an assignment of a value to every variable in a given set of variables, as defined in the QPS framework.

In practice, an outcome is a concrete alternative (a decision, product, agreement etc.). The major difference is that not all attributes may be known. In a sense, such alternatives can be seen as partial outcomes (or sets of outcomes that share some attributes). Even though not all attributes may be specified beforehand, it is often possible to derive the values of some of the unspecified variables using background information. For example, if it is not specified whether someone speaks the language for a given holiday option, such information may be inferred if it is known that the destination is Barcelona which is in Spain, where the language is Spanish, Juan will accompany Anne, and he speaks Spanish.

In this section we introduce an extension of the argumentation framework in which it is possible to reason with such background knowledge. To this end, we extend the language and add one more inference scheme. This extension makes the system more flexible in case of incomplete specifications. If some attributes remain unknown even with reasoning with background knowledge, the argumentation framework still works correctly, it will just infer less preferences.

4.1 Language

Background knowledge is expressed using a set of predicates $\mathcal{P}_K$ which may differ per application domain. Atoms built with these predicates may also be negated (strong negation). Furthermore, a new construct is added to the input language: (defeasible) *rules* that consist of a set of (possibly weakly negated) antecedents and a consequent (the same kind of rules are used in [6]).

Definition 10. (Language) The input language is defined as follows.
$atom_{in}$ ::= $p\,(t_1, \dots, t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{in}$
$atom_K$ ::= $p\,(t_1, \dots, t_n)$ where p is an n-ary predicate $\in \mathcal{P}_K$
$literal_{in}$ ::= $atom_{in} \mid atom_K \mid \neg atom_K$
$rule$ ::= $literal_{in}, \dots, literal_{in}, \sim literal_{in}, \dots, \sim literal_{in} \Rightarrow literal_{in}$
$formula_{in}$ ::= $literal_{in} \mid rule$
The full language is defined as follows.
$atom_{out}$ ::= $p\,(t_1, \dots, t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{out}$
$literal$ ::= $literal_{in} \mid atom_{out}$
$formula$ ::= $literal \mid \sim literal \mid rule$

Contradictory formulas Adding strong negation to the language also adds an additional way in which two formulas can be contradictory.

- A and $\neg A$ contradict each other.

Example 5. *Anne's criteria for a holiday are the same as before, but the information that she has about her options is different. The values of the variables* `sl`, `su` *and* `bb` *on which her preferences are based are not specified. Instead, for every outcome she only knows who of her friends is going with her (*`fr`*): Juan (*`j`*) or Mario (*`m`*), and the destination (*`de`*): Barcelona (*`b`*) or Rome (*`r`*). Besides, she has some relevant background information. All of this is specified in the following knowledge base.*

Some facts from the background knowledge:

```
in(b,spain)                 in(r,italy)
mediterranean(spain)        mediterranean(italy)
language(spain,spanish)     language(italy,italian)
speaks(j,spanish)           speaks(m,italian)
beenTo(b)
```

Some rules from the background knowledge:

```
val(O,fr,X), val(O,de,C), in(C,Cn), language(Cn,L), speaks(X,L) => val(O,sl,true)
~val(O,sl,true) => val(O,sl,false)
val(O,de,C), in(C,Cn), mediterranean(Cn), ~val(O,su,false) => val(O,su,true)
val(O,de,C), beenTo(C) => val(O,bb,true)
~val(O,bb,true) => val(O,bb,false)
```

Facts about some of the possible outcomes:

```
val(o1,fr,j)     val(o2,fr,j)     val(o3,fr,m)     val(o4,fr,m)
val(o1,de,b)     val(o2,de,r)     val(o3,de,b)     val(o4,de,r)
```

Information about the preferences:

```
lc(c5)          cc(c4)         sc(c1,sp)     valpref(c1,true,false)
sub(c5,c3)      sub(c4,c1)     sc(c2,s)      valpref(c2,true,false)
sub(c5,c4)      sub(c4,c2)     sc(c3,n)      valpref(c3,false,true)
prior(c5,c3,c4)
```

4.2 Inferences

Defeasible modus ponens This inference rule applies a rule $L_1, \ldots, L_k, \sim L_l, \ldots, \sim L_m$ => L_n: when all its antecedents hold, the consequent is concluded.

$$\frac{L_1, \ldots, L_k, \sim L_l, \ldots, \sim L_m \;\texttt{=>}\; L_n \quad L_1 \quad \ldots \quad L_k \quad \sim L_l \quad \ldots \quad \sim L_m}{L_n}\; DMP$$

Note the difference between a rule in the language and an inference rule. Defeasible modus ponens is an inference rule that applies a rule from the language. We reserve inference rules for domain-independent inferences, and provide the possibility to specify domain-specific rules in the language. Instead of possible undercutters of an inference rule, it is possible to have weakly negated antecedents for the same purpose.

Example 6. *Below are some of the arguments that can be built with the knowledge base from Example 5. The values for the variables* `su` *and* `bb` *can be derived in a similar way.*

$$\frac{r \quad \texttt{val(o1,fr,j)} \quad \texttt{val(o1,de,b)} \quad \texttt{in(b,spain)} \quad \texttt{lang(spain,spanish)} \quad \texttt{speaks(j,spanish)}}{\texttt{val(o1,sl,true)}}$$

where r is `val(O,fr,X), val(O,de,C), in(C,Cn), lang(Cn,L), speaks(X,L) => val(O,sl,true).`

$$\frac{\sim\texttt{val(O,sl,true) => val(O,sl,false)} \quad \overline{\sim\texttt{val(o2,sl,true)}}}{\texttt{val(o2,sl,false)}}$$

The argument deriving a preference for `o1` *over* `o2` *by criterion* `c5` *is the same as in Example 4, except that* `val(o2,bb,false)` *and* `val(o1,bb,true)` *are derived instead of taken directly from the knowledge base (for reasons of space, the argument is cut in three).*

$$\frac{\texttt{lc(c5)} \quad \texttt{sub(c5,c3)} \quad \dfrac{\texttt{sc(c3,bb)} \quad A \quad B \quad \texttt{valpref(c3,false,true)}}{\texttt{pref(c3,o2,o1)}}}{\texttt{pref(c5,o2,o1)}}$$

$$A : \frac{\sim\texttt{val(O,bb,true) => val(O,bb,false)} \quad \sim\texttt{val(o2,bb,true)}}{\texttt{val(o2,bb,false)}}$$

$$B : \frac{\texttt{val(O,de,C), beenTo(C) => val(O,bb,true)} \quad \texttt{val(o1,de,b)} \quad \texttt{beenTo(b)}}{\texttt{val(o1,bb,true)}}$$

5 Conclusion

In this paper we presented an argumentation framework for representing and reasoning about qualitative multi-criteria preferences. We showed that this argumentation framework models the preferences as defined by qualitative preference systems. Qualitative preference systems use both cardinality and lexicographic ordering to combine multiple criteria, which are ultimately based on the attributes of the outcomes. In an extension of the base argumentation framework we added the means to reason with background knowledge, which adds expressivity and flexibility in case of incomplete specifications.

Argumentation about preferences has been studied extensively in the context of *decision making* [7, 8]. The aim of decision making is to choose an action to perform. The quality of an action is determined by how well its consequences satisfy certain criteria. For example, [8] present an approach in which arguments of various strengths in favour of and against a decision are compared. However, it is a two-step process in which argumentation is used only for epistemic reasoning. Also in [9, 10], preferences are based on arguments, but not themselves derived using argumentation. In our approach, we combine reasoning about knowledge, criteria and preferences between outcomes in a single argumentation framework.

Within the context of argumentation, an approach that is related to criteria is *value-based argumentation* [11, 12]. Values are used in the sense of ‘fundamental social or personal goods that are desirable in themselves’ [12], and are used as the basis for persuasive argument in practical reasoning. A value can be seen as a binary criterion that is satisfied if the value is promoted. In value-based argumentation, arguments are associated with values that they promote. Values are ordered according to importance to a particular audience. An argument only defeats another argument if it attacks it and the value promoted by the attacked argument is not more important than the value promoted by the attacker. In this framework, every argument is associated with only one value, while in many cases there are multiple values or interests at stake. [13] define so-called *value-specification argumentation frameworks*, in which arguments can support multiple values, and preference statements about values can be given. However, the preference between arguments is not derived from the preference between the values promoted by the arguments. Besides, there is no guarantee that a value-specification argumentation framework is consistent, i.e., some sets of preference statements do not correspond to a preference ordering on arguments.

In value-based argumentation, we cannot argue *about* what values are promoted by the arguments or the ordering of values; this mapping and ordering are supposed to be given. But these might well be the conclusion of reasoning, and might be defeasible. Therefore, it would be natural to include this information at the object level. [14] describe some argument schemes regarding the influence of certain perspectives on values. However, for the aggregation of multiple values, they assume a given order on sets of values, whereas we want to derive such an order from an order on individual values.

In our future work we would like to look into the possibilities that the presented framework offers to not only derive missing information about the attributes of outcomes, but also information about e.g. the criteria that are used and their preferences between attribute values, or priority between subcriteria. This would be especially useful when modelling other agents’ preferences, e.g. the opponent in negotiation or someone you have to make a joint decision with. Often, another person’s preferences are not (completely) known, but some of them may be inferred by default.

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. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. Journal of Automated Reasoning **29** (2002) 125–169
2. Kakas, A., Moraïtis, P.: Argumentation based decision making for autonomous agents. In: Autonomous Agents and Multiagent Systems (AAMAS 2003). (2003) 883–890
3. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. Artificial Intelligence **77** (1995) 321–357
4. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation **1**(2) (2010) 93–124
5. Vreeswijk, G.A.W.: Abstract argumentation systems. Artificial Intelligence **90**(1-2) (1997) 225–279
6. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics **7** (1997) 25–75
7. Ouerdane, W., Maudet, N., Tsoukiàs, A.: Argumentation theory and decision aiding. In Ehrgott, M., Figueira, J.R., Greco, S., eds.: New Trends in Multiple Criteria Decision Analysis. Springer Verlag (2010)
8. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence **173**(3-4) (2009) 413–436
9. Bonnefon, J.F., Fargier, H.: Comparing sets of positive and negative arguments: Empirical assessment of seven qualitative rules. In: 17th European Conference on Artificial Intelligence (ECAI 2006), IOS Press (2006) 16–20
10. Dubois, D., Fargier, H., Bonnefon, J.F.: On the qualitative comparison of decisions having positive and negative features. Journal of Artificial Intelligence Research **32** (2008) 385–417
11. Bench-Capon, T.J.M.: Persuasion in practical argument using value based argumentation frameworks. Journal of Logic and Computation **13**(3) (2003) 429–448
12. Bench-Capon, T., Atkinson, K.: Abstract argumentation and values. In Rahwan, I., Simari, G.R., eds.: Argumentation in Artificial Intelligence. Springer (2009) 45–64
13. Kaci, S., van der Torre, L.: Preference-based argumentation: Arguments supporting multiple values. International Journal of Approximate Reasoning **48**(3) (2008) 730–751
14. van der Weide, T., Dignum, F., Meyer, J.J., Prakken, H., Vreeswijk, G.: Practical reasoning using values: Giving meaning to values. In: 6th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009). (2009)