Qualitative multi-criteria preference representation and reasoning

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 4 december 2012 om 15.00 uur
door

Wietske Marijn VISSER

Master of Science in Cognitive Artifcial Intelligence
geboren te Roden

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. C.M. Jonker

Copromotor: Dr. K.V. Hindriks

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. C.M. Jonker,	Technische Universiteit Delft, promotor
Dr. K.V. Hindriks,	Technische Universiteit Delft, copromotor
Prof. dr. L.W.N. van der Torre,	University of Luxembourg
Prof. dr. mr. H. Prakken,	Rijksuniversiteit Groningen & Universiteit Utrecht
Prof. dr. ir. G.J.P.M. Houben,	Technische Universiteit Delft
Prof. dr. Y.-H. Tan,	Technische Universiteit Delft
Dr. U. Endriss,	Universiteit van Amsterdam

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.

SIKS Dissertation Series No. 2012-32
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 978-94-6186-078-1
Gedrukt door Wöhrmann Print Service.
Omslagontwerp door Jeltje Visser.

Contents

Chapter 1

Introduction

1.1 The Pocket Negotiator

Julia felt pleased with herself. She had just signed the contract for the new job she was about to start in a couple of weeks. A job contract that was very satisfactory indeed! When she thought back to the dicussion of the contract details of her previous job, she could still feel the disappointment. How different it had been this time. Back then, she had been totally unprepared. Of course, she had had some idea of the salary she wanted, but of course her boss didn't agree and she had had to settle for a compromise. She had never really felt satisfied about that. This time she had been much better prepared. Her friend Michael had been urging her to try his latest gadget: the Pocket Negotiator. She had been hesitant at first, but decided to give it a try. The device had guided her through all the steps needed to prepare for the negotiation. She had been forced to think about what she really wanted and the machine had suggested things that she would never have thought of by herself. It had also advised her to think about the wishes of her new employer and ways for her to meet them. And just now during the meeting itself, the Pocket Negotiator had helped her analyse the offers made by the HR manager, and suggested some counteroffers. The HR manager had been impressed by her arguments and original proposals. In the end, both had happily signed the contract. This was definitely a good start to her new job!

The research reported on in this thesis is part of a larger research project that aims to develop a negotiation support system called the Pocket Negotiator. This thesis focuses on the question how such a system can represent and reason about a user's preferences between the possible outcomes of a negotiation. The Pocket Negotiator project was originally motivated as follows [68].

> 'Negotiation is a complex emotional decision-making process aiming to reach an agreement to exchange goods or services. Although a daily activity, few people are effective negotiators. Existing support systems

> make a significant improvement if the negotiation space is well-understood, because computers can better cope with the computational complexity. However, the negotiation space can only be properly developed if the human parties jointly explore their interests. The inherent semantic problem and the emotional issues involved make that negotiation cannot be handled by artificial intelligence alone, and a human-machine collaborative system is required.
>
> [We] will develop a new type of human-machine collaborative system that combines the strengths of both and reduces the weaknesses. Fundamental in these systems will be that user and machine explicitly share a generic task model. Furthermore, such systems are to support humans in coping with emotions and moods in human-human interactions. For this purpose we will contribute new concepts, methods and techniques. For integrative bargaining we will develop such a system, called a Pocket Negotiator, to collaborate with human negotiators. The Pocket Negotiator will handle computational complexity issues, and provide bidding and interaction advice, the user will handle background knowledge and interaction with the opponent negotiator.
>
> The Pocket Negotiator will enhance the negotiation skills and performance of the user by increasing the user's capacity for exploration of the negotiation space, reducing the cognitive task load, preventing mental errors, and improving win-win outcomes. We will devise a negotiation model that matches human cognitive representations of negotiation, and develop methods and tools to support humans in coping with emotions. Two negotiation domains, labour agreements and real estate acquisition, with associated experts provide the development ground for the Pocket Negotiator. We will validate the techniques and tools in training situations, and realistic experiments.'

From this description we would like to highlight the following points. First, the project concerns negotiation support, which is characterised by *collaboration* between a human negotiator and a support system (software agent). This topic is strongly related to automated negotiation, although there are some differences. Automated negotiation concerns autonomous agents, who, although they may act on a human user's behalf, have the authority to propose, reject and accept bids and commit to agreements on their own. In negotiation support, agents may advise their users on the actions to take, but it is the user who retains responsibility for actions and their consequences. Still, a negotiation support system's strategy will be similar to the strategy of an autonomously negotiating agent; the support system should suggest the action that it would take himself if it were autonomous. In addition, a support system should be able to *explain* its suggestions in order for the user to understand why this action is the best.

Second, the negotiation support system should be able to help in *real-world* negotiations such as buying a house or car, or negotiating about employment conditions.

Although people often think that negotiation is just bargaining over a single issue (usually the price), enlarging the negotiation space can lead to better deals. For example, a deal where you get a good value on some issue that is important to you but less important to your opponent, and your opponent gets a good value on an issue that is important for him but less so for you, is better than a deal where a compromise has been made on both issues. So it is important to take all possible issues into account simultaneously during a negotiation. This means that *outcomes* (possible bids) are complex entities with a specified value for every issue on the table. Since real-world negotiations often involve many issues, each of which can have many different values, the *outcome space* is typically very large.

Third, the goal of a negotiation support system is to help a human negotiator to reach a better deal in negotiation. The quality of a deal is determined for a large part by the user's personal *preferences*, so a negotiation support system should take these into account. Although the satisfaction of a person with the result of a negotiation might also be influenced by other aspects, such as the process of the negotiation, the atmosphere and emotions during the negotiation, or the state of the relationship with the opponent, we will focus here on satisfaction with the deal itself. In real-world negotiations, there is often no clear-cut border between acceptable and unacceptable deals. Rather, deals are considered more or less preferred by either participant of a negotiation on some scale of preference. Since a negotiation support system supports a human user, it should have a *model* of this user's preferences. While some preferences are (almost) universal (e.g. wanting to pay as little as possible), most are subjective and have to be *elicited* from the user. Although most current negotiation support systems use numerical measures such as utility to represent preferences, such quantitative preferences are hard to specify for human users, and therefore hard to elicit. Therefore, it would be more natural to model the user's preferences in a *qualitative* way. Moreover, due to the exponential size of the outcome space in real-world, multi-issue negotiations, it is not feasible to specify a preference ordering directly. Therefore, we aim to represent the preferences in a more compact way by using *multiple evaluation criteria* that influence preference and deriving an overall preference among outcomes by aggregating them. This approach is called multi-criteria preference representation.

In this thesis, we focus on the representation of, and reasoning about, such qualitative, multi-criteria preferences over a complex outcome space. Other research within the Pocket Negotiator project has focused on social acceptance of negotiation support systems [102], the design and evaluation of interfaces for preference construction [100] and value reflection [101], smart bidding strategies in complex negotiation domains [117, 12, 13, 14, 11], the development of an explanation facility based on shared mental models between system and user [77, 79, 78], emotions in negotiation [35, 39] and negotiation training in virtual reality [37, 36].

In the remainder of this chapter, we first give some background on (automated) negotiation in order to illustrate the exact context (Section 1.2). Then in Section 1.3 we discuss the field of preference handling. Finally, in Section 1.4 we specify the research objectives and give an overview of the structure of this thesis.

1.2 Negotiation

Rosenschein and Zlotkin [113] identify three main components in negotiation: the space of possible deals, the negotiation process, and the negotiation strategy.

1.2.1 Space of possible deals

To what kind of agreements can the agents come? The price for a given good or service is a different kind of agreement than a job contract, which in turn is different from a plan for joint action. To bring some structure in the wide variety of domains of negotiation, Rosenschein and Zlotkin [113] identify a three-tier hierarchy of negotiation domains: task oriented, state oriented and worth oriented domains. Task oriented domains are a subset of state oriented domains, which in turn are a subset of worth oriented domains.

In a *task oriented domain*, each agent is assigned a set of tasks that it can carry out by itself (it has all the resources and capabilities needed and other agents cannot interfere). Negotiation then is about redistributing tasks among the agents to everyone's benefit. In a *state oriented domain*, agents have a specific goal to reach. Agents aim to move from an initial state to a state where their goal is satisfied. There may be multiple goal states, and multiple ways to reach a goal state. Also, goals of different agents may conflict, and agents may not have all resources or capabilities to reach their goal. So negotiation is about which state is to be reached, and about the allocation of (scarce) resources. In contrast to state oriented domains, where an agent's goal will either be satisfied or it will not (goal satisfaction is binary), in *worth oriented domains* it is possible to specify a degree of satisfaction: agents assign a worth or value to every possible outcome that captures its desirability. In worth oriented domains it is possible for agents with opposite desires to reach a compromise.

The aim of the Pocket Negotiator project is to provide negotiation support in a wide variety of (real-world) domains, e.g. employment conditions and real estate. As discussed above, outomes in such negotiations are complex entities that have specified values for a set of issues. Such outcomes are not split into acceptable and unacceptable ones, but are rather ordered according to their relative preference. This means that we have to model the highest level of negotiation domains: worth oriented domains.

1.2.2 Negotiation process

Given a set of possible deals, what is the process that agents can use to converge to agreement on a single deal? What are the rules that specify how consensus will be reached? A *negotiation protocol* is a set of rules that govern the interaction. It specifies things such as who can take part in the negotiation, which actions are allowed and what their consequences are, when the negotiation ends, and how agreements are enforced. A commonly used protocol in automated negotiation is the alternating offers protocol. This is a protocol for bilateral (two-party) negotiation, in which the participants take turns in making an offer. The only actions that are allowed are making an offer, accepting an offer from the opponent, or leaving the negotiation.

In real-world human-human negotiations, the negotiation protocol is not very strict. Besides offers, other information may be exchanged, and parties may try to persuade each other. A branch in automated negotiation that aims to formalize this kind of interaction is *argumentation-based negotiation* [5, 3, 6, 21, 67, 73, 81, 88, 89, 97, 96, 110, 114, 116]. Here, the exchange of offers is extended with other possible moves. Argumentation-based negotiation is commonly seen as a dialogue in which multiple locutions or speech acts are possible. Some commonly used locutions are propose, accept, reject, assert, challenge, justify, promise, and threat. One of the advantages of argumentation-based argumentation is that information and arguments about the negotiators' preferences can be exchanged. Having an accurate model of the opponent's preferences can improve the quality of the negotiation outcome. One way to acquire such information is just asking for it. Argumentation-based negotiation provides the means for such communication. Moreover, through the use of argumentation, negotiators can justify their negotiation stance and influence the other's negotiation stance. This can lead to more efficient negotiations.

1.2.3 Negotiation strategy

Given a set of possible deals and a negotiation process, what strategy should an individual agent adopt while participating in the process? A distinction that is often made in the literature on negotiation is between approaches that are based on game theory, heuristics, and argumentation [110]. The *game-theoretic approach*, e.g. [113], studies how negotiation protocols and strategies can be defined such that they satisfy certain desirable properties, such as termination (every negotiation will end), efficiency (there is no deal that is better than the agreement that is reached), and equilibrium (there is no incentive to play a different strategy). Although game-theoretic approaches are very powerful, Rahwan *et al.* [110] argue that they have some significant limitations because they make some strong assumptions, such as complete and correct information and rationality of agents, which cannot be made in real life. Negotiators often keep their preferences to themselves, so that preferences of the opponent are not (completely) known. Also, nothing is known about the opponent's rationality. *Heuristic-based approaches*, e.g. [53, 69], do not make the strong assumptions that the game-theoretic approach makes. Strategies in this approach are based on certain rules of thumb that apply in most cases. However, they cannot be proven to lead to an optimal solution. Therefore, this approach relies on experimental testing, for example through simulation of negotiations with various parameters. In *argumentation-based negotiation*, an agent's strategy should not only determine what offers he will make, but also what information to share, what arguments to use to try to persuade the opponent, and how to react to the opponent's arguments. As such arguments are likely to include preference information, an agent in argumentation-based negotiation should at least be able to reason about his own and the opponent's preferences.

Another distinction is between agents that keep a model of the opponent (preferences, strategy, trustworthiness, etc.) and those that do not. It is argued that opponent modelling increases the effectiveness and efficiency of negotiation [117]. We focus here on modelling the opponent's (qualitative) preferences. Such an opponent model may be based on some general assumptions (e.g. sellers prefer higher prices). An agent can then try to find out more about the opponent's preferences by exchanging arguments. For example, the agent might ask explicitly for the opponent's preferences or underlying interests (e.g. [55]). Finally, an agent could try to change the opponent's preferences. This may be done in a friendly manner, by pointing out some information that the opponent hasn't considered, or making a promise. A less friendly approach (but maybe more effective) is making a threat. This is where persuasive argumentation comes into play (e.g. [116, 81, 112]). Finding out and changing preferences through the exchange of arguments has been studied in the field of interest-based negotiation [106, 111, 108] and some studies have been performed as to the effectiveness of this approach [55, 98, 109]. However, this work only concerns the task oriented domain. To our knowledge, no such study has yet been performed in worth oriented negotiation domains.

1.3 Preference handling

Preferences are studied and applied in many different contexts. We briefly mention a few examples here that give a feeling for the wide variety of situations in which preferences play a role. In *decision making* (e.g. [76, 28, 49, 51, 95, 7]), it is the task of the decision maker to select an action that is the best (or at least good enough, depending on the setting) according to some preference model. Special attention is given to multi-criteria decision making (MCDM) and decision making under uncertainty (DMU). Decision making typically involves experts, who have to make important decisions, which means that some time can be spent in the construction of an accurate preference model. In *recommender systems* (e.g. [115, 91, 120]), a system recommends one or more items to a user according to the system's model of the user's preferences. This approach can be applied in e-commerce, where relatively little time can be spent on preference elicitation, but on the other hand data about many different users are available. Preferences also play a major role in the field of *social choice* (or collective decision making) (e.g. [132, 83, 84, 42, 85]). The task here is to choose an option based on the preferences of multiple parties. Often this is accomplished by some kind of voting mechanism. In *negotiation support*, finding the best outcome for the user is not enough. Since the need for negotiation originates from a conflict of interests, the user is unlikely to obtain his most preferred outcome. Rather, a deal is struck that satisfies both negotiation partners to an acceptable degree. This means that it is necessary to construct a preference model that represents the preferences of the user between at least all acceptable states. Moreover, it is beneficial to also have such a model of the opponent's preferences.

The field of preference handling (for an overview, see [31, 47, 70]) consists of several related aspects. First, any system working with a user's preference has to have an accurate *representation* or *model* of those preferences (see Section 1.3.1). Second, it needs to be able to *reason* with that model (see Section 1.3.2). Third, in order for a model to be constructed, it has to be *elicited* from the user (see Section 1.3.3). These aspects are not independent, but the choices made for one influence the choices that can be made in the others. Chevaleyre *et al.* [42] mention five objectives or desirable properties that should be considered when making such choices. First, the language should be *expressive*. This is measured in the types of relations that can be expressed. For example, the preference relations expressed by a CP-net [29] always have a particular lattice-type structure, and the leximin ordering in the prioritized goals approach is always total. Second, the language should be *succinct*. Every ordering can be expressed by explicitly listing it, but in any reasonably-sized outcome space this is infeasible. Coste-Marquis *et al.* [44] investigate the expressive power and relative succinctness of some propositional preference representation languages. Third, the language should have *low computational complexity* if it is to be used in practice. Fourth, it is an advantage if the language is *elicitation-friendly*. Related to this is the last objective, of *cognitive relevance*, i.e. that the preference representation resembles the way humans think about preferences.

1.3.1 Representation

A first broad distinction that can be made in preference representation is that between quantitative and qualitative approaches. In quantitative approaches, each alternative is associated with a numeric value, the *utility* of that option. Approaches differ in how such a value is computed. For example, in decision making under uncertainty, the expected utility of a decision can be used, which is based on the utilities of its possible outcomes and their probabilities. In multi-criteria decision making, the overall utility is based on the degree of satisfaction of each of a set of evaluation criteria. In negotiation, a deal or outcome generally consists of multiple issues. For a complete deal, negotiators have to agree on the value for every issue. The satisfaction of a negotiator with a possible deal depends on his preferences over the values of the various issues. This is not straightforward. A commonly used approach (also used in the automated negotiating agents competition (ANAC, [14, 11])), is based on weighted utilities: every issue has an associated weight indicating its importance and every possible value of every issue has a certain utility. The overall utility of a possible deal is then determined by the sum of weighted utilities of the issues' values. This is known as a factored value function. This approach makes the assumption of preferential independence, as discussed in depth by Keeney and Raiffa [76]. If a user's preferences are conditional, which is often the case, it will not be able to reflect the user's true preferences. To overcome the limitiations of this approach, a generalization has been proposed, called generalized additive independence (GAI) value functions [31, 54, 15]. This language is fully general and can represent any value function. A GAI value function can also be represented graphically in a GAI-network [59]. This approach is very powerful, but it makes the strong assumption that numeric utilities and weights are available, which is not always the case.

In qualitative approaches, preference is not defined as a value function but as a binary relation between alternative outcomes. Such a relation is commonly defined as at least a preorder (i.e. a reflexive and transitive relation). In some approaches it is also assumed to be total and/or antisymmetric. If a preference relation is a total preorder it can also be represented by assigning a rank (or utility) to every outcome, but this is not possible if the relation is partial (i.e. if there are preferentially incomparable outcomes).

The question is how to represent such a relation. One possibility is to use explicit comparison statements, e.g. 'I like this car better than that car'. Unfortunately, this does not provide guidance for ordering new outcomes that are not mentioned in any statement, and providing a full preference relation quickly becomes infeasible in any real-life domain. Therefore, preference relations are commonly represented in a more compact way. Here, the structure of outcomes is advantageous. Outcomes are defined as assignments of values to a set of attributes or variables (if all variables are Boolean, outcomes correspond to propositional models). This gives the option of generalizing preference statements, e.g. 'I like red cars better than black cars'. Such statements relate to one specific evaluation criterion. Approaches that use multiple evaluation criteria to determine preference differ both in the types of criteria that are used and in the way (the preferences induced by) the criteria are aggregated.

One type of criterion, which is especially used in the case where outcomes are represented as propositional models, is *goals*. Although there is no consensus on the exact definition of a goal, in this context it can be seen as some desired proposition that is either satisfied or not. Every goal splits the outcome space in two: a set of outcomes that satisfies the goal and a set of outcomes that does not. Another type, that can be seen as a generalization of a goal, is to let an evaluation criterion assign a *level of satisfaction* to every outcome. Usually the scales of satisfaction are shared among all criteria (i.e. criteria are commensurate). Sometimes the scales are bipolar, i.e. they distinguish between negative, neutral and positive degrees. Finally, it is possible for every criterion to specify its own *preference relation* on the set of outcomes, which can be any preorder.

There are more considerations that distinguish the different approaches. One is the question whether criteria can be conditional or not, e.g. whether it is possible to specify that in summer, I prefer to go on holiday in Crete, while in winter I prefer the Alps. Another question is what 'framing' can be used by the criteria, i.e. on which variables they can be based. Sometimes (e.g. in CP-nets [29]), criteria are defined directly on the variables whose values define the outcomes. In other cases it is also possible to define the criteria on more abstract, derived concepts. In that case it would be possible to specify for example that I like to go on holiday to some place where I can either sunbathe on the beach or ski (thus expressing the same preference as in the conditional case, but in a more abstract way).

If multiple criteria play a role, the overall preference is determined by an aggregation of those criteria. This can be done in several different ways. Possibly the best-known approach is the *ceteris paribus* ('all other things being equal') interpretation of preference statements [22, 138]. Here, the statement 'I like red cars better than black cars' is interpreted in such a way that a red car is preferred to a black car if both cars are the same on all other relevant aspects. What exactly the other relevant aspects are depends on the specific approach. They can be the values of the other variables, or the satisfaction (level) of the other criteria. A well-known framework, called CP-nets [29], combines the ceteris paribus semantics with conditional preferences. Extensions of this approach that incorporate relative importance [32] and stronger conditional statements [136] have been proposed as well.

If goals are used as criteria, obviously the outcomes that satisfy all goals are the most preferred. In real-life applications, however, such outcomes may not be available. In that case other ways are needed to determine preference between the available options. A simple approach is to count the number of goals that are satisfied. The more goals an outcome satisfies, the more preferred it is. This approach can be refined by assigning a weight to every goal that indicates its importance [41, 119, 118]. Instead of assigning weights, the importance of a goal can also be indicated qualitatively, as is done in the prioritized goals approach (e.g. [33]). Here, every goal has an associated rank (multiple goals can have the same rank). Different strategies to obtain a preference ordering can be applied, such as the leximin and discrimin orderings. For example, the leximin strategy prefers one outcome over another if there is a rank where the first satisfies more goals than the second, and for every more important rank, they satisfy the same number of goals.

Sometimes a compensation or trade-off between the satisfaction and the importance of criteria can be made. This requires commensurability of the scales used for measuring the satisfaction of all criteria and their relative importance.

Finally, an operator that can combine any arbitrary preference relations induced by criteria is the lexicographic rule. Here, criteria are ordered according to priority by a strict partial order (a transitive and asymmetric relation). The lexicographic rule weakly prefers one outcome over another if for every criterion, either this criterion also weakly prefers the first outcome over the second, or there is another criterion with a higher priority that strictly prefers the first over the second. Andréka *et al.* [8] prove that this rule is in fact the only operator for combining arbitrary preference relations that satisfies all of the desired properties IBUT (independence of irrelevant alternatives, based on preferences only, unanimity with abstentions, and preservation of transitivity).

1.3.2 Reasoning

Once a preference model is there, it should be able to answer some queries. There are two kinds of queries that are used often [29]: *outcome optimization* (finding the best outcome) and *preferential comparison* (determining the preference relation between two given outcomes). Outcome optimization is needed in decision making; only one decision can be made and it should be the best possible one. Also, a recommendation system works best if it recommends the most preferred options. On the other hand, in negotiation, finding the best outcome for one party is not enough. Typically, negotiating parties have conflicting preferences so that one party's most preferred outcome is not likely to be accepted. In this case we need a preference ordering of some or all relevant possible deals, on which a negotiation strategy can be based. To this end, the preferential comparison query is used. A third kind of query is to determine for a given outcome whether it is good enough (satisficing). This approach is useful when time or resources are limited. There are different interpretations of what it means to be good enough. One is that the outcome should be better than some reference outcome, such as the current situation. In this case, the query would correspond to a preferential comparison query between the given outcome and the reference outcome.

In order to answer such preference queries, the system has to be able to *reason* with the preference model. The algorithms for this depend on the chosen preference representation. In general, there is a trade-off between the expressivity of the representation format and the complexity of the query algorithms. For example, a well-known framework that defines both a representation language and algorithms that answer queries is CP-nets [29]. In an acyclic CP-net, outcome optimization is easy. In fact, the optimal outcome is not selected from a set of given outcomes, but rather 'constructed' by choosing the most preferred value for every variable, where variables whose preference is unconditional are assigned first, and the dependent variables are assigned after that. This approach is possible because of the assumptions that the CP-net framework makes, resulting in a preference ordering that has the form of a lattice with a unique optimal outcome which may be constructed by assigning the most preferred value to every variable. Comparison queries in the CP-net framework are answered by constructing so-called 'improving flipping sequences' (a sequence of outcomes where each outcome differs from the previous one in the value of exactly one variable, and outcomes are increasingly preferred). This algorithm is also only applicable due to the specific structure of a preference relation induced by a CP-net, but in contrast to outcome optimization it is not easy but PSPACE-complete in general.

One option to reason about preferences is to use argumentation. Following the seminal work of Dung [52], formal argumentation has grown to be a core study within artificial intelligence [18]. Besides providing a reasoning mechanism for single agents (see e.g. [104]), argumentation is also applied in communication between multiple agents [90, 87, 107].

Using argumentation for preference modelling has several advantages. First, since argumentation is a form of defeasible reasoning, it is equipped to deal with incomplete and inconsistent information. This is often the case in preference reasoning. Second, argumentation is modular, in the sense that arguments are not proofs, but single reasons for a given statement. That is, although arguments may have the structure of a logical proof, their acceptability is determined by their interaction with other arguments. The overall conclusion can only be drawn when the relations among arguments are clear. Adding new reasons will not affect the existing arguments, but it might influence the overall conclusion. This corresponds nicely to the use of multiple criteria to determine the overall preference. For example, if one house is bigger than another, this might be a reason or argument for preferring the first. If the first house is also more expensive, this might in turn be a reason for preferring the second. The overall preference for one house over the other can only be decided if all known reasons are compared. Argumentation itself does not pin down the strategy to be used; that depends on the underlying logical language, which can be defined at will. Third, argumentation is clear, because arguments can be built in a step-by-step fashion, using inference rules. By defining the inference rules in a natural way, the structure of an argument will reflect the reasoning steps that are made. This means that arguments that are formed for reasoning can also be used to explain the (current) preference model to the human user, or to support statements in a dialogue with other agents. This can be useful in e.g. recommendation or bidding support (especially in argumentation-based negotiation).

In existing approaches, argumentation is mostly applied in the context of decision making (e.g. [72, 9, 7, 94, 133]). In such approaches, arguments are built in favour of or against certain decisions. Through the interplay of attacking arguments, one decision should eventually be chosen as the best. Although the arguments are typically based on evaluation criteria, this approach does not really reason about preferences. The conclusions of arguments typically involve statements about a single decision, and the attack between arguments advocating different decisions is implicit, due to the fact that only one option can be chosen. In contrast, reasoning about preferences themselves would involve direct comparisons between options.

Another approach to reasoning about preferences is to use modal logic. Especially when outcomes are seen as propositional models, the step to the possible worlds of modal logic is an intuitive one. Modal logics have been used in several ways to represent and reason about preferences. For example, Boutilier [28] presents a logic with a possible worlds semantics to model qualitative probabilities and preferences that can represent (defeasible) conditional preferences. Van Benthem *et al.* [22] present a modal logic that formalizes the ceteris paribus preferences as initiated by Von Wright [138]. Liu [86] and Girard [58] both use modal logic to model preference change.

1.3.3 Elicitation

Before a preference model can be used in practice in a system, it has to be constructed or instantiated. To this end, preferences have to be elicited from the human user. Since preference elicitation is likely to be an iterative process, an existing preference model also needs to be updateable. We can distinguish several ways of constructing and updating a preference model.

First, a default preference model can be derived from data about the user. This is particularly useful if many data are available about different users and their preferences in the current domain. This approach is similar to collaborative filtering, where a customer's preferences are estimated from the preferences of other customers. If the domain is known, and preferences are known for many users, then this step is useful. One advantage is that it saves time because not all criteria have to be added manually. If the required data are not available, this step has to be skipped. In any case, further elicitation steps are necessary, since although this step may provide a good starting point, it will not be accurate enough and has to be personalized further.

Second, criteria and priorities can be inserted and/or updated manually. This requires that the user understands the used representation framework, or the preference model can be intuitively displayed, and that the user knows his own preferences well. If this is the case, the process is clear and the resulting preference model is accurate. Disadvantages are that it is only suitable for (nearly) expert users, and can take a lot of time.

Third, a preference model can be constructed or updated by incorporating information that is acquired by asking the user particular questions, or by observing the user's behaviour (see e.g. [43] for CP-nets, [27] for conditionally lexicographic preference relations). The aim of this approach is to reduce the user's cognitive load, but still acquire an accurate preference model.

1.4 Thesis overview

The main research objective of this thesis is to develop a framework for the representation of, and reasoning about a user's preferences in the context of a negotiation support system. Above, we have motivated our assumption that such preferences are qualitative and based on multiple criteria, and range over a complex domain of outcomes. The specific research questions that we address in this thesis are the following.

1. *How can argumentation be used to reason about qualitative multi-criteria preferences?*

2a. *How can qualitative multi-criteria preferences be derived when information about the outcomes is incomplete?*

2b. *How can qualitative multi-criteria preferences be derived when information about the outcomes is uncertain?*

3. *What kind of attributes should be chosen as criteria?*

4a. *How can a general framework for the representation of qualitative multi-criteria preferences over multi-attribute domains be defined?*

4b. *How expressive is the proposed framework?*

4c. *How expressive are binary goals as criteria?*

5a. *How can a preference model be explained?*

5b. *How can explanations of preference provided by a user be used by a system to update the preference model?*

6. *How can modal logic be used to reason about qualitative preferences and the relations between preference orderings?*

Every chapter in this thesis deals with a specific aspect of modelling qualitative, multi-criteria preferences, thus covering one or more research questions. All chapters are shortly introduced below. The general structure of the thesis is shown in Figure 1.1. Except for this Introduction and the Conclusion, every chapter is a copy of an article. Except for layout and some minor corrections, the articles are left unchanged. This means that there is a certain amount of unavoidable overlap, but also that every chapter is self-contained and can be read independently from the others.

Chapter 2: Incomplete and uncertain information

This chapter first presents an argumentation-based framework for the modelling of, and reasoning about qualitative multi-criteria preferences. This *basic framework* uses a simple definition of objects (outcomes) and preferences between them. Objects are defined as value assignments to a set of attributes (variables) which are all binary (Boolean). For preference, a version of the lexicographic ordering is used where the criteria are the same as the attributes that define the objects, and the importance (priority) between them is a total preorder (this definition is the same as the leximin or # ordering used in prioritized goals [44, 33]). An *argumentation framework*, including a logical language, a set of inference rules, and a definition of the defeat relation between arguments, is then defined to reason about preferences between objects (research question 1).

The second part of this chapter considers the question of how to reason about preferences when only *incomplete information* about the objects is available (research question 2a). We first discuss some naive strategies of dealing with preferences between objects for which it is not known for every attribute whether it is true or false. From the limitations of these strategies, we identify two desired properties for strategies handling preferences based on incomplete information: decisiveness and safety. We then propose an adequate strategy that is both decisive and safe, based on the notion of least and most preferred completions of objects. This definition generalizes the simple preference definition used in the first part of the chapter: if all information is complete, it results in the same preferences. Finally, the argumentation framework defined in the first part is extended to incorporate this strategy for handling incomplete information (research question 1).

The third part of this chapter deals with the case of *uncertain information* about objects (research question 2b). It first explores how uncertain (defeasible) information can be represented ordinally using certainty levels (degrees of belief), and defines an epistemic argumentation framework to reason about uncertain facts. Then it considers how to reason about preferences between objects for which the truth or falsehood of attributes is uncertain. After discussing some purely qualitative strategies, we define a compensatory strategy and a safer compensatory strategy, which are based on the notion of subjective probability. The safer compensatory strategy generalizes both the compensatory strategy and the decisive and safe strategy for handling incomplete information from the second part of the chapter. Both strategies are also incorporated into the basic argumentation framework for modelling preferences that

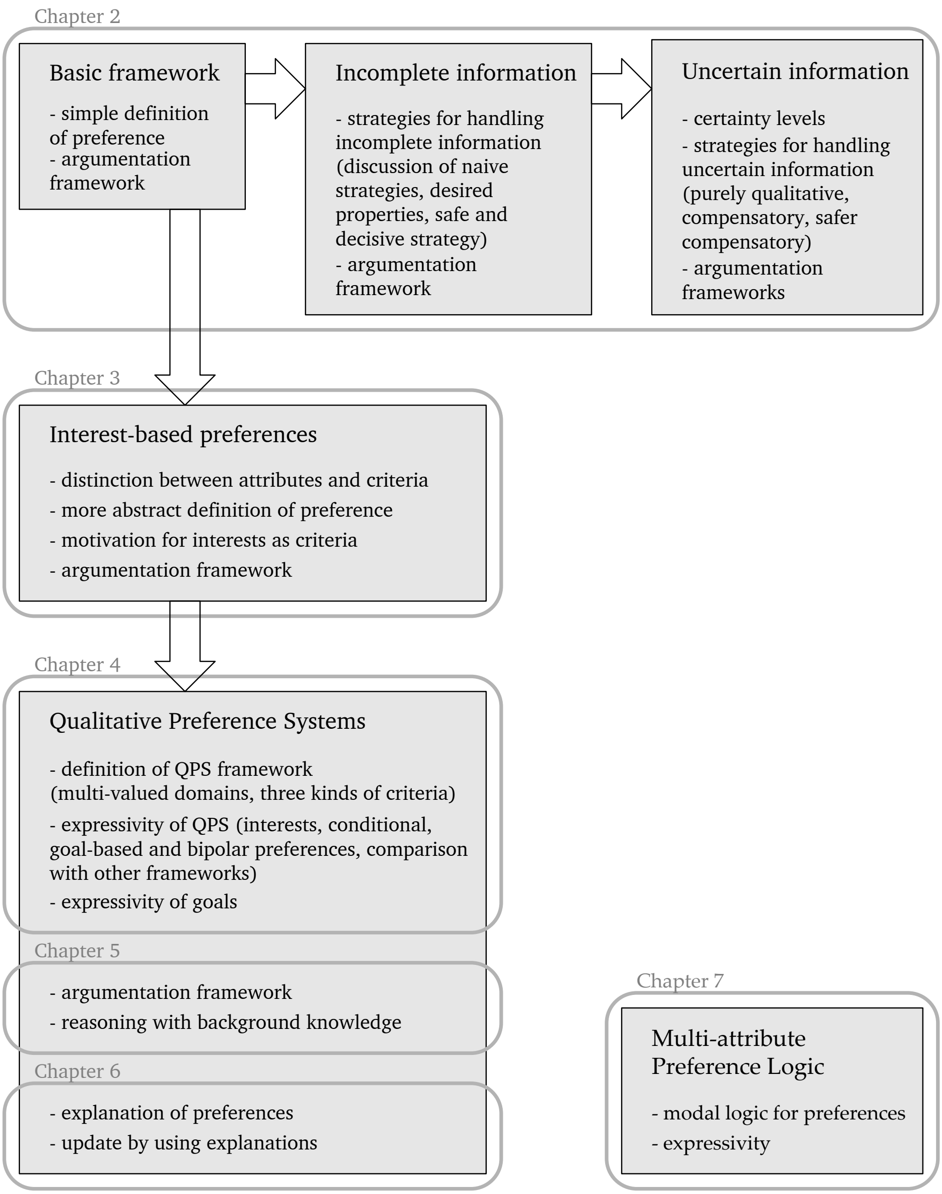


Figure 1.1: Thesis overview

was presented before (research question 1).

Chapter 2 is a copy of [127], which is based on two previous publications [124, 125].

[127] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Argumentation-based qualitative preference modelling with incomplete and uncertain information. *Group Decision and Negotiation*, 21(1):99–127, 2012.

[124] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Argumentation-based preference modelling with incomplete information. In *CLIMA X*, volume 6214 of *Lecture Notes in Artificial Intelligence*, pages 141–157. 2010.

[125] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. An argumentation framework for deriving qualitative risk sensitive preferences. In *Modern Approaches in Applied Intelligence*, volume 6704 of *Lecture Notes in Computer Science*, pages 556–565, 2011.

Chapter 3: Interest-based preferences

This chapter addresses the question what kind of attributes should be chosen as criteria (research question 3). It argues that instead of issues (the attributes that define negotiation outcomes), the negotiators' underlying *interests* should be chosen, especially if the issues are not preferentially independent. Using interests as criteria is more flexible than modelling conditional preferences, and provides a better explanation of the derived preferences.

While this chapter still considers binary (Boolean) attributes, the definition of preference is more abstract compared to the one used in the basic framework in Chapter 2. Here, criteria can also be derived attributes, and the importance between them can be any preorder, thus generalizing both the lexicographic variant used in Chapter 2 and ceteris paribus preference. As in the previous chapter, an argumentation framework is defined that models the proposed preference definition, this time taking interests explicitly into account (research question 1).

Chapter 3 is a copy of [126].

[126] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Interest-based preference reasoning. In *3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, pages 79–88, 2011.

Chapter 4: Qualitative Preference Systems

This chapter presents a general framework for the representation of qualitative, multi-criteria preferences, called *Qualitative Preference Systems* (QPS) (research question 4a). The model is more general than the ones presented in the previous chapters, in which attributes and criteria were assumed to be binary. Here, outcomes are defined as value assignments to a set of variables which can have arbitrary domains. Three types of criteria are defined. Simple criteria derive a preference relation over outcomes from a preference relation on the values of a single variable. Multiple criteria can be combined in a cardinality criterion, which is based on counting the number of criteria that support a preference, or in a lexicographic criterion, which is based on priority. Together, all used criteria form a layered structure called a criterion tree.

After the definition of the QPS framework, the chapter considers the *expressivity* of the framework (research question 4b). It shows that QPS can model conditional preferences and underlying interests, goal-based preferences, and bipolar preferences. It also compares the QPS framework in detail with two other well-known approaches, Logical Preference Description language [33] and CP-nets [29], and gives a translation from both languages into QPS.

Finally, the chapter considers the expressivity of *goals* (binary criteria), even when the domains of the variables that define the outcomes are not Boolean themselves (research question 4c). It shows that any QPS (including simple criteria ranging over multi-valued variables) can be translated to an equivalent and just as succinct goal-based QPS where all simple criteria have been replaced by goals. Moreover, it shows that goal-based QPSs allow more fine-grained updates of the criterion tree because goals relating to different variables can be interleaved.

Chapter 4 is currently submitted for publication in a journal [123]. This article is based on two previous publications [121, 130]. [122] is an extended abstract of [121].

[123] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. Qualitative Preference Systems: A framework for qualitative multi-criteria preferences. Submitted.

[121] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. A framework for qualitative multi-criteria preferences. In *4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, pages 243–248, 2012.

[130] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Goal-based qualitative preference systems. In *10th International Workshop on Declarative Agent Languages and Technologies (DALT 2012)*, 2012.

[122] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. A framework for qualitative multi-criteria preferences: Extended abstract. In *24th Benelux Conference on Artificial Intelligence (BNAIC 2012)*, 2012.

Chapter 5: Argumentation framework for QPS

This chapter presents an *argumentation framework* to reason about preferences expressed in the QPS framework (research question 1). It defines a logical language, a set of inference rules, and a defeat relation. It shows that this argumentation framework models a QPS if the input is a knowledge base containing all information about the outcomes and the criteria. Finally, an extension of the argumentation framework is proposed in which it is possible to reason with *background knowledge* to derive information about the values of variables by default. This is useful when outcomes are not completely specified but the unspecified values are dependent on other variables.

Chapter 5 is a copy of [128].

[128] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. An argumentation framework for qualitative multi-criteria preferences. In *Theory and Applications of Formal Argumentation (TAFA 2011)*, volume 7132 of *Lecture Notes in Artificial Intelligence*, pages 85–98. 2012.

Chapter 6: Explaining QPS

The topic of this chapter is the *explanation* of preference models. Especially for systems that support a human user, it is important that their reasoning, and hence their models of the user's preferences, can be explained in a natural way (research question 5a). This chapter proposes to use the structure of a QPS criterion tree to generate explanations for the resulting preferences between outcomes. It uses the intuition that preferences can be explained by the criteria that are deciding in the overall preference. Explanations are proposed for every kind of preference by every type of criterion.

Next, the chapter considers how explanations given by the user can be used to *update* the current preference model as maintained by the system (research question 5b). Detailed interaction diagrams are provided that specify how the system should react to an explanation, given by the user, of a preference that does not follow from the current model. There are basically two possibilities: to ask the user a follow-up question or to update the preference model. In the latter case, the updated preference model will not only support the same preference as stated by the user, but also generate the same explanation for it.

Chapter 6 is a copy of [129].

[129] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Explaining qualitative preference models. In *6th Multidisciplinary Workshop on Advances in Preference Handling (M-PREF 2012)*, 2012.

Chapter 7: Multi-Attribute Preference Logic

This chapter takes a different approach than the other chapters. It introduces a *modal logic*, called Multi-Attribute Preference Logic (MPL), that provides a language for expressing several strategies to qualitatively derive a preference between objects (outcomes) from property (attribute) rankings (research question 6). Objects here are defined as specific sets of possible worlds (propositional models) that share the same truth assignments. Preferences are derived from a set of desired properties (propositional formulas) that are ranked according to importance. Three different strategies from the literature on prioritized goals [44, 33] to derive preferences from property rankings are modelled. The additional value of the logic is that it is possible to reason about these different preference orderings within the logic. This means we cannot only reason about which objects are preferred according to a certain ordering, but also about the relation between different orderings.

Chapter 7 is a copy of [65]. Two extended abstracts of this article have been published as well [64, 63].

[65] Koen V. Hindriks, Wietske Visser, and Catholijn M. Jonker. Multi-attribute preference logic. In N. Desai, A. Liu, and M. Winikoff, editors, *PRIMA 2010*, volume 7057 of *Lecture Notes in Artificial Intelligence*, pages 181–195. 2012.

[64] Koen Hindriks, Catholijn Jonker, and Wietske Visser. Reasoning about multi-attribute preferences (extended abstract). In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1147–1148, 2009.

[63] Koen Hindriks, Catholijn Jonker, and Wietske Visser. Reasoning about multi-attribute preferences. In *21st Benelux Conference on Artificial Intelligence (BNAIC 2009)*, pages 319–320, 2009.

Chapter 8: Conclusion

The final chapter presents some general conclusions and discusses possible directions for future research.

Chapter 2

Argumentation-based qualitative preference modelling with incomplete and uncertain information

Abstract This paper presents an argumentation-based framework for the modelling of, and automated reasoning about multi-attribute preferences of a qualitative nature. The framework presents preferences according to the lexicographic ordering that is well-understood by humans. Preferences are derived in part from knowledge. Knowledge, however, may be incomplete or uncertain. The main contribution of the paper is that it shows how to reason about preferences when only incomplete or uncertain information is available. We propose a strategy that allows reasoning with incomplete information and discuss a number of strategies to handle uncertain information. It is shown how to extend the basic framework for modelling preferences to incorporate these strategies.

2.1 Introduction

Our introduction of an argumentation-based framework for modelling qualitative multi-attribute preferences under incomplete or uncertain information is motivated by research into negotiation support systems. 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 provide a better correspondence with the way preferences are expressed 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 [49] 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-attribute* 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 [76], 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. On the other hand, also several qualitative approaches have been proposed [33, 34, 44, 86].

Next, a user's preferences and knowledge about the world may be incomplete, uncertain, inconsistent and/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 *incomplete* or *uncertain*, but we will address other types of incompleteness, uncertainty, inconsistency and change in future.

The topic is related to decision making under uncertainty (e.g. [51, 28]). In DMU, the aim is to find the best decision in case of uncertainty about the current state of the world, and hence about the outcomes of decisions. Our approach is more general and can be applied in different contexts; we compare the preference between abstract 'objects', which could be states of the world (as in decision making), but also e.g. products, contracts, holiday arrangements, or houses. Also, the best option may not always be available (e.g. in negotiation, you typically have to find a compromise) so that also the preference between non-optimal solutions is important.

One of the challenges of reasoning about preferences is their multi-attribute nature. There are several distinct notions: importance of attributes, degree of satisfaction of attributes, and degree of belief of facts. In some approaches, (some of) these measures are assumed to be commensurate (e.g. Amgoud and Prade [7] and Keeney and Raiffa's classical utility theory [76]), others (including this paper) suppose *non-commensurability*. In this paper we focus on the case where it is not completely certain which attributes the objects have (there are different degrees of belief), combined with relative importance of attributes. We leave the degree of satisfaction of attributes for future work. Dubois *et al.* [50] present several multi-attribute preference ordering rules, but do not take uncertainty into account. Bonet and Geffner [24] present a qualitative model for decision making with plausibility measures of input situations, but they treat plausible and likely beliefs equally. Amgoud and Prade [7] present an argument-based approach to multi-criteria decision making, but assume that the knowledge base is consistent, fully certain and complete.

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 [18]. 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 or when the available information is not certain. Because of its non-monotonic nature, argumentation is useful for handling inconsistent, incomplete and uncertain information. Although a lot of work has been done on argumentation-based negotiation (for a comprehensive review, see [110]), 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 [4], 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 2.2, we introduce qualitative multi-attribute preferences, in particular the lexicographic preference ordering. In Section 2.3 we start by modelling this ordering for reasoning with complete and certain information in an argumentation framework. Then we proceed and extend this framework in such a way that it can also handle incomplete information. In Section 2.4, we propose a strategy (based on the lexicographic ordering) with some desired properties to derive object preferences in the case of incomplete information. In Section 2.5 this strategy is subsequently incorporated into the argumentation framework. In Section 2.6 we discuss the situation where information about objects is uncertain and introduce an epistemic argumentation framework to reason with such uncertain information. Section 2.7 presents concrete, qualitative preference strategies that provide different ways for handling uncertain information. Section 2.8 concludes the paper.

2.2 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. [33]). 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 $\trianglerighteq$. This relation is not required to be antisymmetric, so two or more attributes can have the same importance. The relation $\trianglerighteq$ 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. Brewka [33] and Coste-Marquis *et al.* [44] 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 defines a total preference relation (contrary to the discrimin ordering) and it is more discriminating than the best-out ordering. Furthermore, the experimental research of Bonnefon and Fargier [26] shows that among several qualitative approaches to order options based on their positive and negative aspects, cardinality-based approaches such as the lexicographic ordering best predict the actual choices made by humans. Since the other orderings are structurally similar to the lexicographic ordering, a similar argumentation framework could be defined for them if desired.

The lexicographic preference ordering first considers the highest importance level. If some object has more attributes on that level than another, the first is preferred.

	large $\triangleq$	*garden* $\triangleq$	*closeToWork* $\triangleright$	*nearShops* $\triangleq$	*quiet* $\triangleright$	*detached*
villa	✓	✓	✗	✗	✗	✓
apartment	✓	✗	✓	✓	✗	✗
cottage	✗	✓	✗	✓	✓	✓

Table 2.1: An example of objects and attributes

If both objects have the same number of attributes on this level, the next importance level is considered, and so on. Two objects are equally preferred if they have the same number of attributes on every importance level. We illustrate the lexicographic preference ordering by means of an example.

Example 2.1. Paul wants to buy a house. According to him, the most important attributes are *large* (e.g. 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*, an *apartment* and a *cottage*. The attributes of these objects are displayed in Table 2.1. In this table, the attributes are ordered in decreasing importance from left to right. $\triangleq$ between attributes indicates equal importance, $\triangleright$ a transition to a lower importance level. A ✓ indicates that an object has the attribute, a ✗ 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 have two of these attributes, while the *cottage* only has 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* has one attribute and the *villa* has none. So the *apartment* is preferred over the *villa*. Note that although the *cottage* has the most attributes in total, it is still the least preferred option because of its bad score at the more important attributes.

Definition 2.1. (Lexicographic preference ordering) Let $\mathcal{P}$ be a set of attributes or goals, and $\trianglerighteq$ a total preorder on $\mathcal{P}$ representing the relative importance among attributes. We write $P \triangleright Q$ for $P \trianglerighteq Q$ and $Q \ntrianglerighteq P$, and $P \triangleq Q$ for $P \trianglerighteq Q$ and $Q \trianglerighteq P$. 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 $|\{P' \mid a$ has P' and $P \triangleq P'\}| > |\{P' \mid b$ has P' and $P \triangleq P'\}|$ and for all $Q \triangleright P$: $|\{Q' \mid a$ has Q' and $Q \triangleq Q'\}| = |\{Q' \mid b$ has Q' and $Q \triangleq Q'\}|$. Object a is *equally preferred* as object b according to the lexicographic ordering if for all P: $|\{P' \mid a$ has P' and $P \triangleq P'\}| = |\{P' \mid b$ has P' and $P \triangleq P'\}|$.

2.3 Basic argumentation framework for preferences

In this section we present an argumentation framework for deriving preferences according to the lexicographic ordering, based on complete and certain information. In later sections we extend this basic framework in order to deal with incomplete and uncertain information.

2.3.1 Abstract AF and semantics

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 [52]. An abstract argumentation framework [52] is a pair $\langle \mathcal{A}, \rightarrow \rangle$ where $\mathcal{A}$ is a set of arguments, and $\rightarrow$ a binary defeat relation (informally, a counterargument relation) on $\mathcal{A}$.

To define which arguments are justified, we use Dung's [52] preferred semantics.

Definition 2.2. (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 $B \rightarrow A$ then $\exists C \in S : C \rightarrow B$. An argument is credulously (sceptically) *justified* w.r.t. preferred semantics if it is in some (all) preferred extension(s).

Informally, a preferred extension is a coherent point of view that can be defended against all its attackers. In case of contradictory information there will be multiple preferred extensions, each advocating one point of view. The contradictory conclusions will be credulously, but not sceptically justified.

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.

2.3.2 Arguments

Arguments are built from formulas of a logical language (see Section 2.3.4), that are chained together using inference steps (see Section 2.3.5). 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 (similar to e.g. Vreeswijk [131]).

Definition 2.3. (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*. `inf` returns the last inference of an argument (the root node), and `conc` returns the conclusion of an argument (the conclusion of its last inference).

Some example arguments will be given in Example 2.3 after the presentation of the specific language and inference schemes that are used to build them.

2.3.3 Defeat

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 Section 2.3.5). 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 2.4. (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.

2.3.4 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 with equal importance as 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 2.5. (Language) Let $\mathcal{P}$ be a set of attribute names with typical elements P,Q, and $\mathcal{O}$ a set of object names with typical elements a,b, and let n be a non-negative integer. The *input language* $\mathcal{L}^{in}$ and full *language* $\mathcal{L}$ are defined as follows.

$$\varphi \in \mathcal{L}^{in} ::= P(a) \mid \neg P(a) \mid P \rhd Q \mid P \triangleq Q$$

$$\psi \in \mathcal{L} ::= \varphi \in \mathcal{L}^{in} \mid pref(a,b) \mid eqpref(a,b) \mid has(a,[P],n)$$

Formulas of this language have the following informal meaning:

$P(a)$	object a has attribute P
$\neg P(a)$	object a does not have attribute P
$P \rhd Q$	attribute P is more important than attribute Q
$P \triangleq Q$	attribute P is equally important as attribute Q
$\mathit{pref}(a, b)$	object a is strictly preferred over object b
$\mathit{eqpref}(a, b)$	object a is equally preferred as object b
$\mathit{has}(a, [P], n)$	object a has n attributes equally important as attribute P (not necessarily including P itself)

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 from $\mathcal{L}^{in}$. 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 $P \rhd Q$, $Q \rhd P$ or $P \triangleq Q$.

Example 2.2. The information from Example 2.1 can be expressed in the form of the following knowledge base that is based on the language $\mathcal{L}^{in}$.

$$\mathit{large} \triangleq \mathit{garden} \triangleq \mathit{closeToWork} \rhd \mathit{nearShops} \triangleq \mathit{quiet} \rhd \mathit{detached}$$

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

2.3.5 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, and has a label. The inferences that can be made are defined by inference schemes. The inference schemes of our framework are listed in Table 2.2. 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* [103], 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 2 can be applied on any subset of the set of attributes of some importance level that an object a has. This means that it is possible to construct an argument that does not

1	$$\frac{}{has(a,[P],0)}\ count(a,[P],\varnothing)$$
2	$$\frac{P_1(a) \quad \ldots \quad P_n(a) \quad P_1 \triangleq \ldots \triangleq P_n}{has(a,[P_1],n)}\ count(a,[P_1],\{P_1,\ldots,P_n\})$$
3	$$\frac{P_1(a) \quad \ldots \quad P_n(a) \quad P_1 \triangleq \ldots \triangleq P_n \triangleq P}{count(a,[P],S \subset \{P_1,\ldots,P_n\}) \text{ is inapplicable}}\ count(a,[P],S)uc$$
4	$$\frac{has(a,[P],n) \quad has(b,[P'],m) \quad P \triangleq P' \quad n > m}{pref(a,b)}\ prefinf(a,b,[P])$$
5	$$\frac{has(a,[Q],n) \quad has(b,[Q'],m) \quad Q \triangleq Q' \triangleright P \quad n \neq m}{prefinf(a,b,[P]) \text{ is inapplicable}}\ prefinf(a,b,[P])uc$$
6	$$\frac{has(a,[P],n) \quad has(b,[P'],m) \quad P \triangleq P' \quad n = m}{eqpref(a,b)}\ eqprefinf(a,b,[P])$$
7	$$\frac{has(a,[Q],n) \quad has(b,[Q'],m) \quad Q \triangleq Q' \ntrianglelefteq P \quad n \neq m}{eqprefinf(a,b,[P]) \text{ is inapplicable}}\ eqprefinf(a,b,[P])uc$$

Table 2.2: Inference schemes for the basic argumentation framework (complete and certain information)

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. 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 2.3. We now illustrate the inference schemes with some arguments that can be made from the knowledge base in Example 2.2. The example arguments are listed in Table 2.3 (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 fact that there is an importance level where the apartment has two attributes and the cottage only one. Argument B illustrates a zero count. Here

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

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

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

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

Table 2.3: Example arguments

a preference for the apartment over the villa is derived, based on the fact that there is an importance level where the apartment has one attribute 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).

2.3.6 Validity

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

Proposition 2.1. Let $\mathcal{A}(KB)$ denote all arguments that can be built from a knowledge base KB. Then there is an argument $A \in \mathcal{A}(KB)$ such that the conclusion of A is $\mathit{pref}(a, b)$ and A is sceptically justified under preferred semantics iff a is preferred over b according to the lexicographic preference ordering (Definition 2.1) given KB.

Proof. Suppose a is preferred over b. This means that there exists an attribute P such that $|\{P' \mid a \text{ has } P' \text{ and } P \triangleq P'\}| > |\{P' \mid b \text{ has } P' \text{ and } P \triangleq P'\}|$ and for all $Q \rhd P$: $|\{Q' \mid a \text{ has } Q' \text{ and } Q \triangleq Q'\}| = |\{Q' \mid b \text{ has } Q' \text{ and } Q \triangleq Q'\}|$. Let $P_1 \ldots P_n$ denote all attributes of equal importance as P such that a has P_i and let $P'_1 \ldots P'_m$ 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: $P_1 \triangleq \ldots \triangleq P_n \triangleq P'_1 \triangleq \ldots P'_m$ and $P_1(a) \ldots P_n(a)$ and $P'_1(b) \ldots P'_m(b)$. 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):

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

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 $P_j \triangleq P$ and $P_j \notin \{P_1, \ldots, P_n\}$ and $P_j(a)$ is the case. However, $P_1 \ldots P_n$ 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 $P_1 \ldots P_n$ (resp. $P'_1 \ldots P'_m$) 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 *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 $Q \rhd P$: $|\{Q' \mid a \text{ has } Q' \text{ and } Q \triangleq Q'\}| = |\{Q' \mid b \text{ has } Q' \text{ and } Q \triangleq Q'\}|$. 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 every preferred extension and hence sceptically justified according to preferred semantics.

Suppose a is not preferred over b. This means that for all attributes P, either $|\{P' \mid a \text{ has } P' \text{ and } P \triangleq P'\}| \leq |\{P' \mid b \text{ has } P' \text{ and } P \triangleq P'\}|$ or there exists an attribute $Q \rhd P$ such that $|\{Q' \mid a \text{ has } Q' \text{ and } Q \triangleq Q'\}| \neq |\{Q' \mid b \text{ has } Q' \text{ and } Q \triangleq Q'\}|$. This means that any argument with conclusion *pref*(a, b) (which has to be of the form above) is either undercut by *count*$(b, [P], S)$*uc* because it uses a non-maximal count, or by *prefinf*$(a, b, [P])$*uc* because there is a more important level where a preference can be derived. This means that any such argument will not be in any preferred extension and hence not sceptically justified under preferred semantics.

The same line of argument can be followed for *eqpref*. □

2.4 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 not always the case. In this section we 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 on attributes that are more important than P, the knowledge whether $P(b)$ or $\neg P(b)$ is the case is irrelevant. But otherwise this information is necessary 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.

2.4.1 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 particular 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

	$P \triangleq Q \triangleq R$		
a	✓	?	✗
b	✗	✓	?
c	?	✗	✓

Table 2.4: Example of intransitive preference with the disregard attribute strategy

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 in Section 2.3 without adaptation would boil down to using a pessimistic strategy: if it is not known whether an object has 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. It 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 have 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.

This strategy has another undesired effect. Consider the situation in Table 2.4. When comparing a and b, this strategy only takes attribute P into account, and concludes a preference of a over b. Similarly, preferences of b over c, and of c over a can be derived. So with this strategy, intransitive preferences can be derived, which is undesired.

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.

2.4.2 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.

2.4.3 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 a.

	P	$\triangleright$ Q	$\triangleright$ R
a	✓	✓	?
b	?	✗	✓

b.

	P	$\triangleright$ Q
a	✓	?
b	?	✓

c.

	P	$\triangleq$ Q
a	✓	?
b	✗	✓

Table 2.5: Examples of objects and attributes with incomplete information

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 2.6. (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 2.5a. Recall our assumption that presence of attributes is preferred over absence. So in the worst case for a, a does not have attribute R. And 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 2.5b. 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 2.5c. 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.

2.5 Argumentation framework for preferences with incomplete information

This section presents how our framework is extended to incorporate the decisive and safe strategy for incomplete information as presented in Section 2.4.3. We first present the changes to the language and then the changes to the inference rules. The defeat definition does not have to change.

2.5.1 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 2.5a. 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 by Prakken and Sartor [105]. 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 $wpref(a, b)$ which express weak preference, just as $pref(a, b)$ and $eqpref(a, b)$ 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.

This leads to the following redefinition of the language.

Definition 2.7. (Language) Let $\mathcal{P}$ be a set of attribute names with typical elements P, Q, and $\mathcal{O}$ 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 *input language* $\mathcal{L}^{in}$ and full *language* $\mathcal{L}$ are defined as follows.

$$\varphi \in \mathcal{L}^{in} ::= P(a) \mid \neg P(a) \mid P \rhd Q \mid P \triangleq Q$$

$$\psi \in \mathcal{L} ::= \varphi \in \mathcal{L}^{in} \mid pref(a^x, b^y) \mid eqpref(a^x, b^y) \mid wpref(a^x, b^y) \mid has(a^x, [P], n) \mid \ \sim \psi$$

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

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

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

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

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

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

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

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

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

8 $\dfrac{}{\sim\varphi}\ asm(\sim\varphi)$ 9 $\dfrac{\varphi}{asm(\sim\varphi) \text{ is inapplicable}}\ asm(\sim\varphi)uc$

10 $\dfrac{pref(a^-,b^+)}{pref(a,b)}$ 11 $\dfrac{eqpref(a^-,b^+) \quad pref(a^+,b^-)}{wpref(a,b)}$

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

Table 2.6: Inference schemes for incomplete information

2.5.2 Inferences

The inference rules of the extended framework are listed in Table 2.6. 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 2.6. 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 is 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 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 2.4. In the case of Table 2.5a, argument A in Table 2.7 can be built. Argument B shows that a weak preference can be inferred in the situation of Table 2.5c.

2.6 Uncertain information

In the last two sections we focused on the situation where some information regarding the presence or absence of attributes for a given outcome is lacking. With the proposed safe and decisive strategy however, it may still be 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. Still, in many situations there is other information available on the basis of which the missing facts can be derived. For example, if the destination country of a certain holiday is not given, but it is specified that the trip will be to Rome, we can infer that the country will be Italy. In this case, the derived fact is completely certain since there is no doubt that Rome is in Italy. In other cases, the derived information may be less than fully certain, e.g. because the applied rule only holds by default and there are some exceptions to it, or because the used facts or applied rule are not certain themselves, e.g. because the source of the information is unreliable. For example, for a holiday to Rome in July we can infer that it will be sunny because that is usually the case. But this conclusion is not completely certain because it may be an exceptionally rainy July in Rome this year. Or we may conclude that the hotel we will stay in will be clean based on the reviews we have read online, but again this conclusion is not certain because we cannot trust the source completely (the hotel itself may have posted fake reviews).

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

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

Table 2.7: Example arguments

The main point we want to make here is that even if missing information can be derived, the acquired facts may have different certainty levels. Such 'degrees of belief' play an important role in the deriviation of preferences. Consider a simple case in which preference is determined by a single attribute P (like before, we assume that presence of an attribute is preferred over absence). If both for outcome a and for outcome b it can be derived that P is present, but for a this conclusion is more certain than for b, it would be rational to prefer a. This is because there is a bigger chance that the information about b is incorrect. The situation gets more complicated in case of multiple attributes. For example, are two certainly true attributes and one certainly false attribute better or worse than three attributes whose truth is not completely certain? In Section 2.7 we present different qualitative strategies to derive preferences in case of more or less certain information. But first we will formalise the concept of certainty levels.

2.6.1 Certainty levels

Different approaches to model degrees of belief can be found in the literature. Among the best-known ones are subjective probability, Dempster-Shafer belief functions, and possibility theory (see [66] for an overview and references). In this paper we take a qualitative approach in which the knowledge base is stratified according to the certainty of the formulas (see also Amgoud *et al.* [1], who use similar certainty levels but apply them to decision making in a different way). Each stratum in the knowledge base corresponds to one level of certainty. Note that in the literature, the notion of certainty levels (or similar notions) is sometimes referred to as preference or priority between formulas. In this paper, we use the term preference only to refer to preference between objects or outcomes, and priority to refer to the relative importance of attributes.

Essentially, a certainty level is the qualitative counterpart of the (subjective) probability of an attribute being true. In the case of the highest certainty level, denoted N, this probability is 1; certainly true information is always true. Similarly, the probability is 0 for the negation of formulas with certainty level N. Also, as is common in the literature on subjective probability, we assume that the subjective probability of truth of a completely uncertain formula (with certainty level 0) is 0.5 (the principle of indifference, [66]). For intermediate certainty levels, an exact probability can not always be given due to the qualitative nature of certainty levels, but intuitively the probability is higher for higher certainty levels.

A knowledge base is $\mathcal{K} = \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_N$ where $\mathcal{K}_1$ contains the knowledge with the lowest certainty, and knowledge in $\mathcal{K}_N$ is fully certain. We assume that $\mathcal{K}_N$ is consistent, but $\mathcal{K}$ may not be. Note that there is no subset $\mathcal{K}_0$; a formula with certainty level 0 is completely uncertain and does not belong in a knowledge base. Every epistemic formula φ in the knowledge base has an associated certainty level l, denoted $\varphi : l$ (i.e. $\varphi \in \mathcal{K}_l$). We will use the notation $\varphi : -l$ to denote that the negation of φ has certainty level l (see inference scheme 6 in Table 2.8). This notation is convenient as it provides a uniform way of expressing the certainty that some attribute is present or absent in an outcome. In this paper we assume that non-epistemic information about the relative importance of desired attributes is fully certain; we leave the situations where this is not the case for future work.

Note that the concept of certainty levels is very general and can also be applied in the cases discussed in the previous sections. If the information is complete and fully certain, there are two certainty levels (true and false): $N = 1$ and level 0 does not occur. The situation with incomplete information is the same except that some information may have certainty level 0 (complete uncertainty). In the generic situation, we have the same three levels, 0 for complete uncertainty and the scale ends N and $-N$ for complete certainty, plus any number of certainty levels in between.

The argumentation framework for deriving preferences based on uncertain information can be considered as consisting of two separate parts: an epistemic part for reasoning about the (uncertain) attributes of outcomes and a preferential part, which contains several approaches to derive preferences from uncertain information. In this paper the main focus is on the preferential part, which will be discussed in Section 2.7. For this part, only the 'output' (justified conclusions) of the epistemic part matters, i.e. which outcomes have which attributes with what certainty. The way in which this information is derived is not important for the preferential part of the framework. However, the preferential part does assume that for every attribute P and every outcome a, there is a single certainty associated with $P(a)$. This means that the epistemic part must resolve conflicts and incompleteness. There are several ways in which to do this in a reasonable way. However, in the remainder of the current section we will give only one possible specification of the epistemic part, and will not discuss all possible alternatives as it is outside the scope of this paper.

2.6.2 Epistemic argumentation framework

In order to derive new facts form other facts in the knowledge base, we introduce a new kind of formula to the input language: rules. A rule is of the form $L_1, \ldots, L_k, \sim L_l, \ldots, \sim L_m \Rightarrow L_n$ where $L_i = P(a)$ or $\neg P(a)$. Its informal reading is: if all of $L_1, \ldots, L_k$ hold, then typically L_n holds, except if one of $L_l, \ldots, L_m$ holds. The same kind of rules was used by Prakken and Sartor [105]. If there are no exceptions, and the rule is fully certain, then it is called a strict rule. Otherwise it is defeasible. Defeasible rules describe what is 'normally' the case. Using this kind of rules can add some information to an incomplete knowledge base. This can be beneficial in situations where a user does not have certain information, and does not have the time or resources to verify information.[1]

Definition 2.8. (Epistemic language) As before, we distinguish a subset of the full language called the input language. A knowledge base can only contain formulas of

[1]For rules to be fully applicable, it would also be required to extend the language with literals that refer to other knowledge than just which outcomes have which attributes, and to specify explicitly which attributes are desired and influence preference. A full discussion of this issue is outside the scope of this paper.

1	$\dfrac{L_1,\ldots,L_m,\sim L_p,\ldots,\sim L_q \Rightarrow L:l \quad L_1:l_1 \quad \ldots \quad L_m:l_m \quad \sim L_p \quad \ldots \quad \sim L_q}{L:\min(l,l_1,\ldots,l_m)}$ *DMP*
2	$\dfrac{}{\sim L}\ asm(\sim L)$
3	$\dfrac{L}{asm(\sim L) \text{ is inapplicable}}\ asm(\sim L)uc$
4	$\dfrac{\sim L \quad \sim \neg L}{L:0}$
5	$\dfrac{L:l \quad \neg L:l}{L:0}$
6	$\dfrac{\neg L:l}{L:-l}$

Table 2.8: Inference schemes for epistemic formulas

the input language; other formulas have to be derived by inference. The epistemic input language $\mathcal{L}_e^{in}$ is defined as follows (where L is a literal ($P(a)$ or $\neg P(a)$, where P is an attribute and a an outcome) and l is a certainty level such that $0 < l \leq N$).

$$\varphi \in \mathcal{L}_e^{in} ::= L:l \mid L,\ldots,L,\sim L,\ldots,\sim L \Rightarrow L:l$$

The full epistemic language $\mathcal{L}_e$ is defined as follows (where L is a literal ($P(a)$ or $\neg P(a)$) and l is a certainty level such that $-N \leq l \leq 0$).

$$\varphi \in \mathcal{L}_e ::= \varphi \in \mathcal{L}_e^{in} \mid \ \sim L \mid L:l$$

To apply a rule, inference scheme 1 in Table 2.8, called defeasible modus ponens, is introduced. The level of certainty of the conclusion is the same as the level of the least certain premise, this is called the weakest link principle (see e.g. [99] for a motivation). The inferences 2 and 3 for weak negation are the same as before.

Now, for any atom φ, we can have any of the following situations.

- Either φ or $\neg\varphi$ (but not both) is in $\mathcal{K}$ or can be derived, with only one level of certainty.
- Neither φ nor $\neg\varphi$ is in $\mathcal{K}$ or can be derived.
- The formula φ occurs in $\mathcal{K}$ or can be derived multiple times with different levels of certainty.
- Both φ and $\neg\varphi$ are in $\mathcal{K}$ or can be derived.

The first situation is the most straightforward case, and this information can be used directly by the preferential part of the framework. The incompleteness in the second situation is a case of complete uncertainty with respect to φ. We use inference scheme 4 in Table 2.8 to derive a certainty level 0 for an atom φ if neither φ nor $\neg\varphi$ can be derived. The other two situations are more complicated, and there are multiple possibilities for handling these cases. For example, if there are multiple arguments concluding φ and/or $\neg\varphi$, one could aggregate these arguments such that arguments with the same conclusion strengthen each other, but such a conclusion is weakened by counter-arguments. This approach is not trivial; see [103] for a discussion of the issues concerning accrual of arguments. For the sake of simplicity, the approach we take here is to make sure that only the conclusion with the highest certainty level will be justified. To this end, the definition of rebuttal would have to be adapted such that an argument only rebuts another argument if their conclusions are each other's negation and the first argument has a higher certainty level than the second. Also, a more certain argument for φ would have to defeat a less certain argument for φ. One remaining issue is what to do with arguments for φ and $\neg\varphi$ with equal certainty. If such arguments rebut each other, there will be multiple preferred extensions, which means less sceptically justified conclusions. One could also argue that this situation is equivalent to the case of complete uncertainty. This can be modelled with inference scheme 5 in Table 2.8.

In order to get the desired results in all cases described above, the definition of defeat has to be slightly changed. First of all, for an argument A to rebut another argument B, the conclusion of A should not only be the negation of the conclusion of B, but it should also be at least as certain. This definition of rebuttal is similar to the ones used in preference-based argumentation [2] and argumentation with defeasible priorities [105]. Next, since we want only the most certain argument for some conclusion to be justified, we introduce a new kind of defeat such that an argument A defeats an argument B if their conclusions are the same but A's conclusion is more certain. The definition of undercut remains unchanged.

Definition 2.9. (Defeat) An argument A *defeats* an argument B if

- $\texttt{conc}(A) = \varphi : l$ and $\texttt{conc}(B) = \neg\varphi : l'$ (*rebuttal*) and $l > l' > 0$, or
- $\texttt{conc}(A) = \varphi : l$ and $\texttt{conc}(B) = \varphi : l'$ and $l > l' > 0$, or
- $\texttt{conc}(A) =$'$\texttt{inf}(B)$ is inapplicable' (*undercut*), or
- A defeats a subargument of B.

In the following, we will assume that the epistemic part of the argumentation framework will resolve conflicts between formulas and their negations with possibly different certainty levels.

2.7 Argumentation framework for preferences with uncertain information

Now that we have introduced a framework for epistemic reasoning with uncertain information, we turn to the question how to derive preferences, if any, from such uncertain information. Different approaches to infer preferences from information with varying degrees of certainty are possible. We discuss several different ones. The strategies we present in this section all apply the lexicographic ordering in the sense that a preference between two objects is determined at the highest importance level of attributes where a preference can be derived. They differ in the way a preference is determined within one importance level.

In the lexicographic ordering, objects are compared w.r.t. their attributes on an importance level. The highest importance level where a preference can be derived determines the overall preference. In the Boolean case, preference within an importance level is determined solely by the number of true attributes of both objects (the number of false attributes can be ignored because it can be computed when the number of true attributes is known). This comparison is relatively easy, since we only have to compare two numbers. In the case of uncertainty, instead of two possible values for an attribute, we have multiple ($2N + 1$), one for each level of certainty. So in this case, we compare two tuples of numbers. Such a comparison can be done in different ways, resulting in different strategies. Abstractly, we can say that the tuples are compared by a 'beats' relation B. If on some importance level object a has m_N certainly true, ..., and m_{-N} certainly false attributes, object b has m'_N certainly true, ..., and m'_{-N} certainly false attributes, and $\langle m_N, \ldots, m_{-N} \rangle >_B \langle m'_N, \ldots, m'_{-N} \rangle$, then object a is preferred over object b on that importance level. The key issue is how to define B.

2.7.1 Purely qualitative strategies

We briefly mention some extreme cases. First, it would be possible to reduce the number of certainty levels to two (N and $-N$) by treating the levels in between either the same way as N (optimistic approach) or the same way as $-N$ (pessimistic approach). This is a generalisation of the optimistic and pessimistic strategies discussed in Section 2.4.1, and the same objections apply. A third option is to treat all positive certainty levels the same way as N and all negative certainty levels the same way as $-N$. This shows great confidence in the correctness of information, but ignores the differences in certainty. Finally, it is possible to treat all certainty levels between $-N$ and N the same way as 0 (this essentially reduces the problem to the case with incomplete information discussed in Section 2.4). This focuses on the uncertainty of the information, but does not take into account that there may be different degrees of uncertainty. Any non-extreme approach should distinguish between different certainty levels.

An obvious strategy is dominance: an outcome a is (weakly) preferred to an outcome b if for all attributes, it is at least as certain that a has it as that b has it. For example, in the situation in Table 2.9a, object a is preferred to object b since its certainty level is at least as high for every attribute. This strategy is clear-cut and quite safe. On the other hand, it is not very decisive: the resulting preference relation is far from complete. Also, it does not take into account what it means for two attributes to be equally important, namely that they are interchangeable. If P and Q have equal importance, it does not matter for preference whether an object has P but not Q, or Q but not P. For example, in the situation in Table 2.9a, objects a and c are incomparable according to dominance, while it would be intuitive to prefer a. To solve this issue, the definition of dominance can be straightforwardly adapted to the following definition of 'ordered dominance'. The attributes within an importance level can be rank-ordered according to the certainty that an object has the attribute (in the case of ties, i.e. multiple attributes having the same certainty for an outcome, consecutive ranks are assigned to them at random). Such an ordering may be different for every object. Now object a is (weakly) preferred to object b if for every rank, it is at least as certain that a has the attribute with that rank in a's ordering as that

	P ≙	Q ≙	R
a	2	1	1
b	1	1	-1
c	1	2	0

a.

	P ≙	Q ≙	R ≙	S ≙	...
a	2	-2	-2	-2	...
b	1	1	1	1	...

b.

	P	≙	Q	▷	R	≙	S	≙	T
a	-1		2		2		2		-1
b	2		-1		1		1		1

c.

	P ≙	Q
a	1	0
b	-1	1

d.

Table 2.9: Examples of objects and attributes with uncertain information

b has the attribute with that rank in b's ordering. This definition results in a preference of a over c in Table 2.9a. This definition captures the intuition behind equal importance of attributes, and again, the derived preferences are intuitive. However, it still lacks in decisiveness, since many objects will be incomparable even though it may be reasonable to prefer one over the other.

In order to define a more decisive strategy, one could consider using the lexicographic ordering on certainty levels. That is, within one importance level, we first count all attributes with certainty N. If one object has more of those than another, the first object is preferred over the second (within this level). If both objects have the same number, we go on to count the attributes with certainty $N - 1$, and so on. When both objects have the same number of attributes on every certainty level, we go on to consider the next importance level. The advantage of this strategy is that it results in a complete preference relation, i.e. two objects are never incomparable. On the other hand, some derived preferences may not be intuitive, since no number of less certain attributes can be valuated higher than a single more certain attribute. Consider for example the situation in Table 2.9b, where $N = 2$. It is known for certain that object a has attribute P and none of the other attributes. Object b has all attributes with certainty 1. The lexicographic strategy will always prefer object a, no matter how many attributes there are in the importance level, even though it would be more intuitive to prefer b when the odds are taken into account. In the next section we propose a strategy that does take the odds into account and allows compensation.

2.7.2 Compensatory strategy

The reason that it would be more intuitive to prefer object b to object a in Table 2.9b as the number of attributes in the importance level increases, is that it becomes more likely that object b will have more attributes than object a. In other words, the expected number of attributes of b, which increases with every attribute with certainty 1 that is added, will be higher than the expected number of attributes of a, which stays 1. In this section we present a strategy for determining preference that is based on the expected number of attributes of each object. As said before, the strategy applies the lexicographic ordering in the sense that a preference between two objects is determined at the highest importance level of attributes where a preference can be derived. Within an importance level, it prefers one object to another if the first has a higher expected number of attributes on that level than the second.

In order to calculate the expected number of attributes that an outcome has on some importance level, we need to know the subjective probability pr associated with each certainty level. Some probabilities may be known, such as $pr(N) = 1$ and $pr(-N) = 0$, others have to be estimated. The probability function has to be monotonic, i.e. $pr(l) > pr(l')$ iff $l > l'$, and $pr(l) = 1 - pr(-l)$. Here we only treat the case for unconditional probabilities. Extending this with conditional probabilities is a straightforward application of well-known techniques and would unnecessarily complicate the presentation of the argumentation framework here. If the probabilities of an outcome having different attributes are independent, given $P_1(a) : l_1, \dots, P_n(a) : l_n$, the expected number of attributes among $P_1, \dots, P_n$ that object a has is given by $\sum_{i=1}^{n} pr(l_i)$.

To incorporate this strategy into the argumentation framework, the interpretation of the formula $has(a, [P], n)$ is changed slightly from 'object a has n attributes equally important as attribute P' to 'object a^x *expectedly* has n attributes equally important as attribute P'. Inference scheme 1 in Table 2.10 takes as premises the attributes of a certain importance level and the certainty levels of an object a having these attributes. It concludes that the expected number of attributes of the object on that importance level is the sum of the probabilities of the certainty levels. Inference scheme 2 is an undercutter that defeats an argument built with scheme 1 if not all attributes at the importance level in question are considered. Note that *all* attributes on the importance level should be included; if some attribute is not present, the certainty level will just be $-N$. Inference scheme 3 infers that an object a is preferred over an object b if the expected number of attributes of a is higher than the expected number of attributes of b on a certain importance level. This scheme is undercut by inference scheme 4 if there is a higher importance level where a and b do not have the same expected number of attributes. Schemes 5 and 6 do the same for equal preference.

Example 2.5. Consider the situation in Table 2.9c. Since both objects have one attribute with certainty 2 and one attribute with certainty -1 on the highest importance level, the preference is determined on the second importance level. Which object is preferred depends on the probabilities of the different certainty levels. Since $N = 2$, we have $pr(2) = 1$ and $pr(-2) = 0$, but the probabilities for the other certainty levels can be subjectively estimated. For example, if we take $pr(1) = 0.75$ and $pr(-1) = 0.25$, the expected number of attributes of object a on the second importance level will be $2 * pr(2) + pr(-1) = 2 * 1 + 0.25 = 2.25$ and the expected number of attributes of object b on the second importance level will be $3 * pr(1) = 3 * 0.75 = 2.25$, and both objects will be equally preferred. This is illustrated with argument A in Ta-

1	$\dfrac{P_1(a):l_1 \quad \ldots \quad P_n(a):l_n \quad P_1 \triangleq \ldots \triangleq P_n}{has(a,[P_1],\sum_{i=1}^{n} pr(l_i))}$	$count(a,[P_1],\{P_1,\ldots,P_n\})$
2	$\dfrac{P_1(a):l_1 \quad \ldots \quad P_n(a):l_n \quad P_1 \triangleq \ldots \triangleq P_n}{count(a,[P_1],S \subset \{P_1,\ldots,P_n\}) \text{ is inapplicable}}$	$count(a,[P_1],S)uc$
3	$\dfrac{has(a,[P],n) \quad has(b,[P],m) \quad n > m}{pref(a,b)}$	$p(a,b,[P])$
4	$\dfrac{has(a,[Q],n) \quad has(b,[Q],m) \quad Q \rhd P \quad n \neq m}{p(a,b,[P]) \text{ is inapplicable}}$	$p(a,b,[P])uc$
5	$\dfrac{has(a,[P],n) \quad has(b,[P],m) \quad n = m}{eqpref(a,b)}$	$eqp(a,b,[P])$
6	$\dfrac{has(a,[Q],n) \quad has(b,[Q],m) \quad Q \rhd P \quad n \neq m}{eqp(a,b,[P]) \text{ is inapplicable}}$	$eqp(a,b,[P])uc$

Table 2.10: Inference schemes for the compensatory strategy for uncertain information

ble 2.11. Other probability estimates lead to other preferences. In argument B, we take $pr(1) = 0.8$ and $pr(-1) = 0.2$, and b is preferred over a. In argument C, we take $pr(1) = 0.7$ and $pr(-1) = 0.3$, and a is preferred over b.

For this strategy we have been specific in assigning probabilities to certainty levels. As can be seen in Example 2.5, small differences in the estimated probabilities can lead to completely different preferences. This makes the strategy decisive (it always infers a preference), but not very safe, since subjective probabilities are not always exactly known and may not be estimated accurately. In the next section we present a safer version of the compensatory strategy that generalises both the compensatory strategy of this section and the safe and decisive strategy for incomplete information presented in Section 2.4.3.

2.7.3 A safer compensatory strategy

The strategy presented here is a generalisation of the compensatory strategy of the previous section, inspired by the strategy for incomplete information presented in Section 2.5. The idea is as follows. Instead of assigning a single probability $pr(l)$ to every certainty level l, we specify a range of probability with a lower bound $pr^-(l)$ and an upper bound $pr^+(l)$. Now we can use the same intuition as before. If the worst case for object a is still preferred over the best case for object b, then a has to be preferred over b (on some importance level). Note that fully certain information still

$$\text{A:}\quad \cfrac{\cfrac{R(a):2 \quad S(a):2 \quad T(a):-1 \quad R \triangleq S \triangleq T}{has(a,[R],2.25)} \qquad \cfrac{R(b):1 \quad S(b):1 \quad T(b):1 \quad R \triangleq S \triangleq T}{has(b,[R],2.25)} \qquad 2.25 = 2.25}{eqpref(a,b)}$$

$$\text{B:}\quad \cfrac{\cfrac{R(a):2 \quad S(a):2 \quad T(a):-1 \quad R \triangleq S \triangleq T}{has(a,[R],2.2)} \qquad \cfrac{R(b):1 \quad S(b):1 \quad T(b):1 \quad R \triangleq S \triangleq T}{has(b,[R],2.4)} \qquad 2.2 < 2.4}{pref(b,a)}$$

$$\text{C:}\quad \cfrac{\cfrac{R(a):2 \quad S(a):2 \quad T(a):-1 \quad R \triangleq S \triangleq T}{has(a,[R],2.3)} \qquad \cfrac{R(b):1 \quad S(b):1 \quad T(b):1 \quad R \triangleq S \triangleq T}{has(b,[R],2.1)} \qquad 2.3 > 2.1}{pref(a,b)}$$

Table 2.11: Some example arguments in the compensatory strategy

1	$\dfrac{P_1(a):l_1 \quad \ldots \quad P_n(a):l_n \quad P_1 \triangleq \ldots \triangleq P_n}{has(a^x, [P_1], \sum_{i=1}^{n} pr^x(l_i))}$	$count(a^x, [P_1], \{P_1, \ldots, P_n\})$
2	$\dfrac{P_1(a):l_1 \quad \ldots \quad P_n(a):l_n \quad P_1 \triangleq \ldots \triangleq P_n}{count(a^x, [P_1], S \subset \{P_1, \ldots, P_n\}) \text{ is inapplicable}}$	$count(a^x, [P_1], S)uc$
3	$\dfrac{has(a^x, [P], n) \quad has(b^y, [P], m) \quad n > m}{pref(a^x, b^y)}$	$p(a^x, b^y, [P])$
4	$\dfrac{has(a^x, [Q], n) \quad has(b^y, [Q], m) \quad Q \rhd P \quad n \neq m}{p(a^x, b^y, [P]) \text{ is inapplicable}}$	$p(a^x, b^y, [P])uc$
5	$\dfrac{has(a^x, [P], n) \quad has(b^y, [P], m) \quad n = m}{eqpref(a^x, b^y)}$	$eqp(a^x, b^y, [P])$
6	$\dfrac{has(a^x, [Q], n) \quad has(b^y, [Q], m) \quad Q \rhd P \quad n \neq m}{eqp(a^x, b^y, [P]) \text{ is inapplicable}}$	$eqp(a^x, b^y, [P])uc$
7	$\dfrac{pref(a^-, b^+)}{pref(a, b)}$	
8	$\dfrac{eqpref(a^-, b^+) \quad pref(a^+, b^-)}{wpref(a, b)}$	
9	$\dfrac{eqpref(a^+, b^-) \quad eqpref(a^-, b^+)}{eqpref(a, b)}$	

Table 2.12: Inference schemes for the safer compensatory strategy for uncertain information

gets a single probability value: $pr^-(-N) = pr^+(-N) = 0$ and $pr^-(N) = pr^+(N) = 1$. The strategy is less decisive than the compensatory strategy presented in the previous section, since it is not always able to derive a preference between two objects. However, it is still more decisive than the dominance-based strategy.

Inference schemes 1 to 6 in Table 2.12 are the same as those for the compensatory strategy in Table 2.10, except that labels ($-$ and $+$) are added to objects and pr^- and pr^+ are used in inference scheme 1. These inference schemes can be used to infer preferences between best and worst cases of objects, similar to best and worst completions in Section 2.5. The inference schemes to infer preferences between objects are exactly the same as before: inference schemes 7 to 9 in Table 2.12 are the same as inference schemes 10 to 12 in Table 2.6.

Example 2.6. Consider the same situation in Table 2.9c again. If we take $pr^+(1) =$

A: $$\dfrac{\dfrac{\dfrac{R(a):2 \quad S(a):2 \quad T(a):-1 \quad R \triangleq S \triangleq T}{has(a^+,[R],2.2)} \quad \dfrac{R(b):1 \quad S(b):1 \quad T(b):1 \quad R \triangleq S \triangleq T}{has(b^-,[R],2.4)} \quad 2.2 < 2.4}{pref(b^-,a^+)}}{pref(b,a)}$$

B: $$\dfrac{\dfrac{\dfrac{R(a):2 \quad S(a):2 \quad T(a):-1 \quad R \triangleq S \triangleq T}{has(a^+,[R],2.25)} \quad \dfrac{R(b):1 \quad S(b):1 \quad T(b):1 \quad R \triangleq S \triangleq T}{has(b^-,[R],2.25)} \quad 2.25 = 2.25}{eqpref(a^+,b^-)} \quad \dfrac{\vdots}{eqpref(a^-,b^+)}}{eqpref(a,b)}$$

C_1: $$\dfrac{\dfrac{P(a):1 \quad Q(a):0 \quad P \triangleq Q}{has(a^-,[P],1)} \quad \dfrac{P(b):-1 \quad Q(b):1 \quad P \triangleq Q}{has(b^+,[P],1)} \quad 1 = 1}{eqpref(a^-,b^+)}$$

C_2: $$\dfrac{\dfrac{P(a):1 \quad Q(a):0 \quad P \triangleq Q}{has(a^+,[P],2)} \quad \dfrac{P(b):-1 \quad Q(b):1 \quad P \triangleq Q}{has(b^-,[P],1)} \quad 2 > 1}{pref(a^+,b^-)}$$

C: $$\dfrac{C_1 \quad C_2}{wpref(a,b)}$$

Table 2.13: Some example arguments in the safer compensatory strategy

0.9, $pr^-(1) = 0.8$, $pr^+(-1) = 0.2$ and $pr^-(-1) = 0.1$, we can build argument A in Table 2.13, concluding that b is preferred over a. If we take $pr^+(1) = 0.8$, $pr^-(1) = 0.7$, $pr^+(-1) = 0.3$ and $pr^-(-1) = 0.2$, no justified arguments concluding a preference between a and b can be constructed.

Proposition 2.2. The safer compensatory strategy presented here generalises the compensatory strategy in Section 2.7.2.

Let $pr^-(l) = pr^+(l)$ for every certainty level l, i.e. the probability range is actually a single probability value. Then for every object a, the expected number of attributes is the same for a^+ and a^- and this strategy coincides with the compensatory strategy in Section 2.7.2.

Example 2.7. Consider the same situation in Table 2.9c again. With $pr^-(1) = pr^+(1) = 0.75$ and $pr^-(-1) = pr^+(-1) = 0.25$, we can construct argument B in Table 2.13, which is analogous to argument A in Table 2.11.

Proposition 2.3. The safer compensatory strategy presented here generalises the safe and decisive strategy in Section 2.4.3.

As said before, the case with incomplete information corresponds to the case with three certainty levels: N and $-N$ (certain presence/absence of attributes) and 0 (unknown). If we take $pr^-(0) = 0$ and $pr^+(0) = 1$, then for every object a, the strategy proposed here counts the number of present and unknown attributes for a^+ and only the certainly present attributes for a^-, and hence coincides with the strategy presented in Section 2.7.

Example 2.8. Consider the situation in Table 2.5c. If we translate this to certainty levels, taking $N = 1$, we get the situation in Table 2.9d. In this case, argument C in Table 2.13 can be built (for reasons of space, this argument's two subarguments are displayed separately). When we compare this argument to argument B in Table 2.7 we see that the conclusions are indeed the same.

2.8 Conclusion

In this paper we have made the following contributions. Approaches based on argumentation 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 or uncertain information. Our approach to the incomplete information case allows to reason about preferences from best- and worst-case perspectives (called completions here), and the consequences for overall preferences. In addition we proposed different ways to reason about preferences in case of uncertain information.

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 by Liu [86].

In the current paper we have focused on the case where we have incompleteness of uncertainty in the epistemic part of the knowledge base (i.e. about the attributes that objects do or do not have). It would be interesting to explore the case where also information about what attributes influence preference and the importance order between them is incomplete or uncertain. This is especially useful when modelling preferences of others, where it is not realistic that all relevant information is available. Other interesting cases are inconsistency in and change of the knowledge that is used to determine preferences.

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'), different degrees of satisfaction of attributes, and preferences based on underlying interests or values. We would also like to look into the relation with e.g. CP-nets [29] and value-based argumentation [71].

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 on how our framework can support a user e.g. by indicating which information is still missing.

Chapter 3

Reasoning about interest-based preferences

Abstract In decision making, negotiation, and other kinds of practical reasoning, it is necessary to model preferences over possible outcomes. Such preferences usually depend on multiple criteria. We argue that the criteria by which outcomes are evaluated should be the satisfaction of a person's underlying interests: the more an outcome satisfies his interests, the more preferred it is. Underlying interests can explain and eliminate conditional preferences. Also, modelling interests will create a better model of human preferences, and can lead to better, more creative deals in negotiation. We present an argumentation framework for reasoning about interest-based preferences. We take a qualitative approach and provide the means to derive both ceteris paribus and lexicographic preferences.

3.1 Introduction

We present an approach to qualitative, multi-criteria preferences that takes underlying interests explicitly into account. Reasoning about interest-based preferences is relevant in decision making, negotiation, and other types of practical reasoning. Since our long-term goal is the development of a negotiation support system, the motivations and examples in this paper are mainly taken from the context of negotiation, but the main ideas apply equally well in other contexts.

The goal of a negotiation support system is to help a human negotiator reach a better deal in negotiation. The quality of a deal is determined for a large part by the user's personal preferences. A deal generally consists of multiple issues. For example, when applying for a new job, some issues are the position, the salary, and the possibility to work part-time. For a complete deal, negotiators have to agree on the value for every issue. The satisfaction of a negotiator with a possible outcome depends on his preferences.

Since the number of possible outcomes is typically very large (exponential in the number of issues), it is not feasible to have the user express his preferences over all possible outcomes directly. It is common to compute or derive preferences over possible outcomes from preferences over the possible values of issues and a weighing or importance ordering of the issues. One of the best-known approaches is multi-criteria utility theory [76], a quantitative approach where preferences are expressed by numeric utilities. Since such quantities are hard for humans to provide, qualitative approaches have been proposed too, e.g. [33]. Our approach is also of a qualitative nature.

In this paper we argue that issues alone are not enough to derive outcome preferences. Instead, we will focus on modelling underlying interests and their relation to issues. There are several reasons for taking interests into account. First, underlying interests can explain and eliminate conditional preferences. Consider the following example. If it rains, I prefer to take my umbrella, but if it doesn't, I prefer not to take it. This is a conditional preference; my preference over taking my umbrella depends on the circumstance of rain. Underlying interests can explain such conditional preferences: I prefer to take my umbrella when it rains because I do not want to get wet, and I prefer not to take it when it's dry because I don't want to carry things unnecessarily. If we take such interests as criteria on which to base preference, we can eliminate conditional preferences entirely. We will get back to this in more detail later. Second, interest-based negotiation is said to lead to better outcomes than position-based negotiation [75, 108]. By understanding one's own and the other party's reasons behind a position and discussing these interests, people are more likely to find more creative options in a negotiation and by that reach a mutually acceptable agreement more easily. A well-known example is that of the two sisters negotiating about the division of an orange. They both want the orange, and end up splitting it in half. Had they known each other's underlying interests, they would have reached a better deal: one sister only needed the peel to make a cake and would gladly have let the other sister have all of the flesh for her juice. Third, thinking about underlying interests is a very natural, human thing to do. Interests are what really matters to people, they are what drive them in their decisions and opinions. Taking underlying interests explicitly into account will result in a better model of human preferences. Such a model is also suited for explanation of the reasoning and advice of a support system.

This last point brings us to the motivation for using argumentation to reason about interest-based preferences. Reasoning by means of arguments is a very human type of reasoning. People often base their decisions on (mental) lists of arguments in favour of and against certain decisions. Therefore argumentation is suitable for explanation of a system's reasoning to a human user. Another advantage of argumentation is that it is a kind of defeasible reasoning. It is able to reason with incomplete, uncertain and contradictory information. Finally, argumentation can be used to (try to) persuade the opponent during negotiation (but this is outside the scope of this paper).

The paper is organised as follows. In Section 3.2 we introduce and discuss the most important concepts that we will use throughout the paper. Then, in Section 3.3, we give an overview of existing approaches to preferences and underlying interests. We give some more details about qualitative multi-criteria preferences in Section 3.4. In Section 3.5 we motivate the explicit modelling of underlying interests, illustrated with examples. Our own approach is presented in Section 3.6. Finally, Section 3.7 concludes the paper.

3.2 Concepts

Before we go on, we will clarify some important concepts that we will use. In negotiation, *issues* are the matters which are under negotiation. An issue is a concrete, negotiable aspect such as monthly salary or number of holidays. Every issue has a set or range of possible values. The value of an issue in a given instance can be objectively determined (e.g. €2400, 30 days). Issues and their possible values typically depend on the domain. Besides the issues under negotiation, there may be other properties of a deal that influence preferences. For example, the location of the company that you are applying to work for can be very important, because it determines the duration of your daily commute, but it is hardly negotiable. Still, such properties are important in negotiation. If, for example, you already got an offer from another company near your home, you will only consider offers that are better taking the location into account.

A *possible outcome* or possible deal has a specified value for every issue. All bids made during a negotiation are possible outcomes. For example, a possible outcome could be a job contract for the position of programmer, with a salary of €3000 gross per month, with 25 holidays, for the duration of one year with the possibility of extension. Any other assignment to the issues would constitute a different outcome. It is the user's preferences over such possible outcomes that we are interested in.

With *criteria* we mean the features on which a preference between outcomes is based. It is common to base preferences directly on the negotiated issues; in that case the issues are the criteria. In this paper we argue that not issues, but underlying interests should be used as criteria.

Many terms are used for what we consider to be *underlying interests*, such as fundamental objectives, values, concerns, goals and desires. In our view, an interest can be any kind of motivation that leads to a preference. Essentially, a preference depends on how well your interests are met in the outcomes to be compared. The degree to which interests are met is influenced by the issues, but there is not necessarily a one-to-one relation between issues and interests. For example, an applicant with childcare responsibilities will have the interest that the children are taken care of after school. This interest can be met by various different issues, for example part-time work, the possibility to work from home, a salary that will cover childcare expenses, etc. One issue may also contribute to multiple interests. Many issues that deal with money do so, because the interests different people have for using the money will be diverse.

3.3 Related work

Existing literature about preferences is abundant and very diverse. In this section we briefly discuss the approaches that are most closely related to our interests.

Interest-based negotiation is discussed by Rahwan *et al.* [108]. However, this approach has a particular view on negotiation as an allocation of indivisible and non-sharable resources. The resources are needed to carry out plans to reach certain goals. Even though the goals can be seen as underlying interests, it is hard to model e.g. negotiation about a job contract as an allocation of resources. Salary might be an allocation of money, but other issues, like position or start date, cannot be translated as easily into resources.

Argumentation about preferences has been studied extensively in the context of *decision making* [1, 7, 93, 94]. 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, Amgoud *et al.* [1] 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. In our approach, we combine reasoning about preferences and knowledge in a single argumentation framework.

Within the context of argumentation, an approach that is related to underlying interests is *value-based argumentation* [17, 16]. Values are used in the sense of 'fundamental social or personal goods that are desirable in themselves' [16], and are used as the basis for persuasive argument in practical reasoning. 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. We will illustrate this with a little example. Consider two job offers a and b. a offers a higher salary, but b offers a better position. We can construct two mutually attacking preference arguments, A: 'I prefer job offer a over job offer b because it has a higher salary', and B: 'I prefer job offer b over job offer a because it has a better position'. In Dung-style argumentation frameworks [52], there is no way to choose between two mutually attacking arguments (unless one is defended and the other is not). In value-based argumentation, we could say that preferring a over b promotes the value of wealth (w), and preferring b over a promotes the value of status (s), and e.g. wealth is considered more important than status. In this case A defeats B, but not the other way around.

In this framework, every argument is associated with only one value, while in many cases there are multiple values or interests at stake. Kaci and Van der Torre [71] 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. Van der Weide *et al.* [134] 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.

3.4 Qualitative multi-criteria preferences

Regardless of whether we take issues or interests as criteria, we need to be able to model multiple criteria. In any realistic setting, preferences are determined by multiple criteria and the interplay between them. Therefore we shortly introduce two well-known approaches to multi-criteria preferences which we will use in our framework.

One approach is *ceteris paribus* ('all else being equal') comparison. One outcome is preferred to another ceteris paribus, if it is better on some criteria and the same on all other criteria. This approach has been widely used since Von Wright [138]. Also Wellman and Doyle [135] derive preferences from sets of goals in a ceteris paribus way. In [29], ceteris paribus comparison is combined with conditional preferences in a graphical preference language called CP-nets. The preference order resulting from ceteris paribus comparison is not complete; an outcome satisfying criterion G but not H cannot be compared to an outcome satisfying H but not G.

Another well-known approach is the *lexicographic* preference ordering (see e.g. [33], where it is denoted #). Here, preferences over outcomes are based on a set of relevant criteria, which are ranked according to their importance. The importance ranking of criteria is defined by a total preorder $\trianglerighteq$, which yields a stratification of the set of criteria into importance levels. Each importance level consists of criteria that are equally important. The lexicographic preference ordering first considers the highest importance level. If some outcome satisfies more criteria on that level than another, then the first is preferred over the second. If two outcomes satisfy the same number of criteria on this level, the next importance level is considered, and so on. Two outcomes are equally preferred if they satisfy the same number of criteria on every level.

We use a slightly more abstract definition of preference that covers both ceteris paribus and lexicographic preferences. Let $\mathcal{C}$ be a set of binary criteria, ordered according to importance by a preorder $\trianglerighteq$. If $P \trianglerighteq Q$ and not $Q \trianglerighteq P$, we say that P is strictly more important than Q and write $P \triangleright Q$. If $P \trianglerighteq Q$ and $Q \trianglerighteq P$, we say that P is equally important as Q and write $P \triangleq Q$. $\mathcal{C}$ can be divided into equivalence classes induced by $\triangleq$, which we call importance levels. An importance level L is said to be more important than L' iff the criteria in L are more important than the criteria in

	high salary	high position	full-time
a	✓	✓	✓
b	✓	✓	✗
c	✓	✗	✓
d	✓	✗	✗
e	✗	✓	✓
f	✗	✓	✗
g	✗	✗	✓
h	✗	✗	✗

a. Issues

	wealth	status	family time
a	✓	✓	✗
b	✓	✓	✓
c	✓	✓	✗
d	✓	✓	✓
e	✗	✓	✗
f	✗	✓	✓
g	✗	✗	✗
h	✗	✗	✓

b. Interests

Table 3.1: Satisfaction of issues and interests

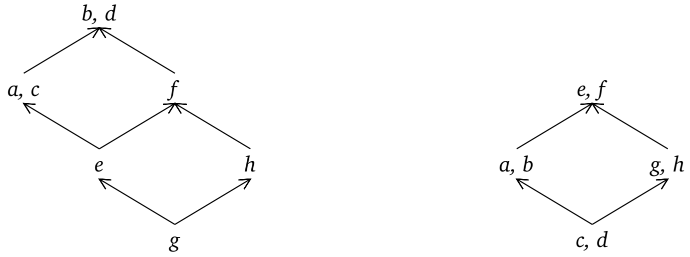
L'. Let $\mathcal{O}$ be a set of outcomes, and sat a function that maps outcomes $a \in \mathcal{O}$ to sets of criteria $\mathcal{C}_a \in 2^{\mathcal{C}}$. If $P \in sat(a)$, we say that a satisfies P.

Definition 3.1. (Preference) An outcome a is *strictly preferred* to another outcome b if it satisfies more criteria on some importance level L, and for any importance level L' on which b satisfies more criteria than a, there is a more important level on which a satisfies more criteria than b. An outcome a is *equally preferred* as another outcome b if both satisfy the same number of criteria on every importance level.

The least specific importance order possible is the identity relation, in which case the importance levels are all singletons and no importance level is more important than any other. In this case, the preference definition is equivalent to ceteris paribus preference (if a is preferred to b ceteris paribus, there are no criteria that b satisfies but a does not). If the importance order is a total preorder, the definition is equivalent to lexicographic preference. In general, the more information about the relative importance of interests is known, the more preferences can be derived. We note that lexicographic preferences subsume ceteris paribus preferences in the sense that if one outcome is preferred to another ceteris paribus, it is also preferred lexicographically, regardless of the importance ordering on criteria.

3.5 Modelling interests

We will illustrate the ideas presented in this paper by means of an example. Mark has applied for a job at a company called Jones. After the first interview, they are ready to discuss the terms of employment. There are three issues on the table: the salary, the position, and whether the job is full-time or part-time. All possible outcomes are listed in Table 3.1a. After some thought, Mark has determined that the interests that are at stake for him are wealth, status, and time with his family. A high position will give status. A high salary will provide both wealth and status. A part-time job will



a. For Mark

b. For Jones

Figure 3.1: Ceteris paribus preference orderings (arrows point towards more preferred outcomes)

give him time to spend with his family. Table 3.1b shows which interests each of the outcomes satisfies.

All information is encoded in a knowledge base, which consists of three parts.

- *Facts* about the properties of the outcomes to be compared. When comparing offers in negotiation, these may be the values for each issue, or any other relevant properties. Facts are supposed to be objectively determined.
- A set of *interests* of a negotiator. Underlying interests are personal and subjective, although they can sometimes be assumed by default. Interests may vary according to importance. If no importance ordering is given, the ceteris paribus principle can be used to derive preferences. The more information about the relative importance of interests is known, the more preferences can be derived. If there is a total preorder of interests according to importance, a complete preference ordering over possible outcomes can be derived using the lexicographic principle.
- *Rules* relating issues and other outcome properties to interests. These rules can be very subjective, e.g. some people consider themselves very wealthy if they earn €3000 gross salary per month, while for others this may be a pittance. Even so, there can still be default rules that apply in general, e.g. that a high salary promotes wealth for the employee. The relation between issues and interests does not have to be one-to-one. There may be multiple issues that can satisfy an interest, some issues may satisfy multiple interests at once, or a combination of issues may be needed to fulfill an interest. As is common in defeasible reasoning, there may be exceptions to rules. For example, one might say that a high position ensures status in general, but this effect is cancelled out if the job is badly paid.

With the inference scheme of defeasible modus ponens (see scheme 1 in Table 3.4), arguments can be constructed that derive statements about what interests are satisfied by possible outcomes, based on their issue values and the rules relating issues to interests. The conclusions from these arguments are summarized in Table

	jacket	pants	shirt
i	b	b	w
j	b	b	r
k	b	w	w
l	b	w	r
m	w	b	w
n	w	b	r
o	w	w	w
p	w	w	r

a. Issues

	good combination
i	✗
j	✓
k	✓
l	✗
m	✓
n	✗
o	✗
p	✓

b. Interests

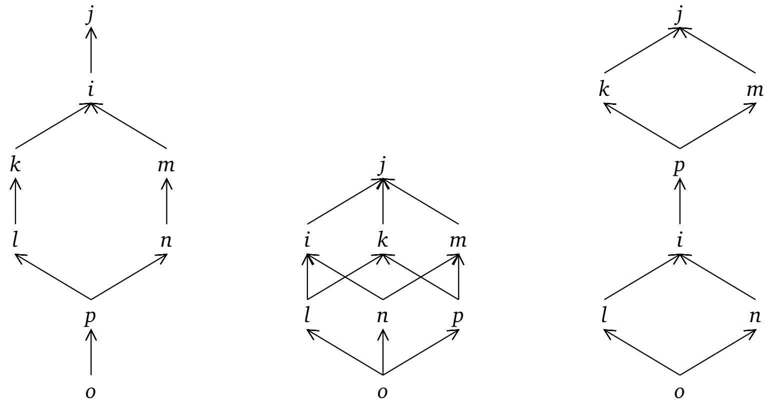
Table 3.2: Outcomes in the evening dress example

3.1b. If we compare the possible outcomes ceteris paribus, we can construct a partial preference order for Mark, with b and d being the most preferred options, and g the least preferred (see Figure 3.1a). This preference order is not complete. To determine Mark's preference between a and c on the one hand and f on the other hand, we need to know whether wealth or family time is more important to him. If wealth is more important, Mark will prefer a or c. If family time is more important, he will prefer f. Similarly, to determine a preference between e and h, we need to know whether status or family time is more important.

The company Jones has two major interests: it needs a manager and it has to cut back on expenses. These interests relate directly (one-to-one) to high position and low salary. The ceteris paribus preference ordering for Jones is displayed in Figure 3.1b.

The added value of interests It may seem that using interests next to issues just introduces an extra layer in reasoning. From the issues and the relations between issues and interests, we derive the interests that are met by outcomes, and from that we derive preferences. Would it not be easier to derive the preferences directly from the issues? We could just state that Jones has the interests of high position and low salary, optionally with an ordering between them, and we would be able to derive Jones' preferences from that. This is because in this case there is a one-to-one relation between interests and issues: every interest is met by exactly one issue, and every (relevant) issue meets exactly one interest.

There are good reasons, however, why this approach is not always a good solution. Consider for example Mark's preferences. A high salary satisfies both wealth and status, and status can be satisfied by either a high salary or a high position. Because of this, the (partial) preference ordering we determined for Mark cannot be defined as a ceteris paribus ordering if the issues are taken as criteria. This is because high position as criterion is dependent on high salary: if the salary is not high, then high position is a distinguishing criterion, but if the salary is high, high position is not



a. Preference graph induced by CP-net

b. Ceteris paribus ordering with interests

c. Ceteris paribus ordering with interests, good combination most important

Figure 3.2: Preference orderings (arrows point towards more preferred outcomes)

relevant anymore, since the only interest that it serves, status, is already satisfied by high salary. So with a fixed set of issues as criteria, ceteris paribus or lexicographic models cannot represent every preference order. In many cases, this can be solved intuitively by taking underlying interests into account.

There are other approaches to deal with this matter. Instead of assuming independence of the criteria, one can also model conditional preferences, where criteria may be dependent on other criteria. A well-known approach to represent conditional preferences is CP-nets [29], which is short for conditional ceteris paribus preference networks. A CP-net is a graph where the nodes are variables (comparable to our notion of issues). Every node is annotated with a conditional preference table, which lists a user's preferences over the possible values of that variable. If such preferences are conditional (dependent on other variables), each condition has a separate entry in the table, and the variables that influence the preference are parent nodes of this variable in the graph. In [29], an example of conditional preference is given regarding an evening dress. A man unconditionally prefers black to white as a colour for both the jacket and the pants. His preference between a white and a red shirt is conditioned on the combination of jacket and pants. If they have the same colour, he prefers a red shirt (for a white shirt will make his outfit too colourless). If they are of different colours, he prefers a white shirt (because a red shirt will make his outfit too flashy). The complete assignments (outcomes in our terminology) are listed in Table 3.2a. The preference graph induced by the CP-net for this example is displayed in Figure 3.2a.

We propose to replace the variables the preferences over which are conditional with underlying interests – the reason for the dependency. In the evening dress example, the underlying interest is that the colours of jacket, pants and shirt make a good combination, which in this case is defined by being neither too colourless nor too flashy. The satisfaction of this interest by the different outcomes is listed in Table 3.2b. The variables jacket and pants are unconditional, so they can remain as criteria. If we take jacket, pants, and good combination as criteria, we can construct the preference graph in Figure 3.2b, using the ceteris paribus principle. The difference with the preferences induced by the CP-net is that in the CP-net case, outcome i is more preferred than k and m, and p is less preferred than l and n, while in the interest-based case they are incomparable. This is due to the fact that in CP-nets, conditional preferences are implicitly considered less important than the preferences on the variables they depend on ([29], p. 145). In fact, if we would specify that both jacket and pants are more important than a good combination, our preference ordering would be the same as in Figure 3.2a. But the interest approach is more flexible; it is possible to specify any (partial) importance ordering on interests. For example, we could also state that a good combination is more important than either the jacket or the pants, which results in the preference ordering in Figure 3.2c. In our opinion, there is no a priori reason to attach more importance to unconditional variables as is done in the CP-net approach.

3.6 Argumentation framework

In this section, we present an argumentation framework (AF) for reasoning about qualitative, interest-based preferences. An abstract AF in the sense of Dung [52] is a pair $\langle \mathcal{A}, \rightarrow \rangle$ where $\mathcal{A}$ is a set of arguments and $\rightarrow$ is a defeat relation (informally, a counterargument relation) among those arguments. To define which arguments are justified, we use Dung's [52] preferred semantics.

Definition 3.2. (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 $B \rightarrow A$ then $\exists C \in S : C \rightarrow B$. An argument is credulously (sceptically) *justified* w.r.t. preferred semantics if it is in some (all) preferred extension(s).

Informally, a preferred extension is a coherent point of view that can be defended against all its attackers. In case of contradictory information, there will be multiple preferred extensions, each advocating one point of view. The contradictory conclusions will be credulously, but not sceptically justified.

We instantiate an abstract AF by specifying the structure of arguments and the defeat relation.

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

$highsal(c)$	$I_M(wealth)$	$highsal(x) \Rightarrow wealth(x)$
$\neg highpos(c)$	$I_M(status)$	$highsal(x) \Rightarrow status(x)$
$full\text{-}time(c)$	$I_M(family)$	$highpos(x) \Rightarrow status(x)$
$\neg highsal(f)$		$\neg full\text{-}time(x) \Rightarrow family(x)$
$highpos(f)$	$I_J(manager)$	$highpos(x) \Rightarrow manager(x)$
$\neg full\text{-}time(f)$	$I_J(cutback)$	$\neg highsal(x) \Rightarrow cutback(x)$

Table 3.3: The knowledge base for the example

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. Vreeswijk [131]).

Definition 3.3. (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*. `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.

Definition 3.4. (Language) Let $\mathcal{P}$ be a set of predicate names with typical elements P, Q; $\mathcal{O}$ a set of outcome names with typical elements a, b; α an audience; and n a non-negative integer. The *input language* $\mathcal{L}_{KB}$ and full *language* $\mathcal{L}$ are defined as follows.

$$\varphi \in \mathcal{L}_{KB} ::= L \mid I_\alpha(P) \mid P \rhd_\alpha Q \mid P \triangleq_\alpha Q \mid L_1, \ldots, L_k, \sim L_l, \ldots, \sim L_m \Rightarrow L_n$$

where $L_i = P(a)$ or $\neg P(a)$.

$$\psi \in \mathcal{L} ::= \varphi \in \mathcal{L}_{KB} \mid \ \sim L \mid sat(a, [P]_\alpha, n) \mid pref_\alpha(a, b) \mid eqpref_\alpha(a, b)$$

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 criteria that outcomes (do not) satisfy, statements about interests of an audience and their importance ordering, and defeasible rules. The knowledge base for the job contract example (the facts restricted to outcomes c and f) is displayed in Table 3.3. Other formulas of the language that are not part of the input language, e.g. expressing a preference between two outcomes, can be derived from a knowledge base using inference steps that build up an argument (such formulas are not allowed in a knowledge base because they might contradict derived statements).

Inferences Table 3.4 shows the inference schemes that are used. The first inference scheme is called defeasible modus ponens. It allows to infer conclusions from defeasible rules. The next two inference rules define the meaning of the weak negation $\sim$. According to inference rule 2, a formula $\sim \varphi$ can always be inferred, but

1	$\dfrac{L_1,\ldots,L_k,\sim L_l,\ldots,\sim L_m \Rightarrow L_n \quad L_1 \quad \ldots \quad L_k \quad \sim L_l \quad \ldots \quad \sim L_m}{L_n}\ DMP$
2	$\dfrac{}{\sim L}\ asm(\sim L)$
3	$\dfrac{L}{asm(\sim L) \text{ is inapplicable}}\ asm(\sim L)uc$
4	$\dfrac{}{sat(a,[P]_\alpha,0)}\ count(a,[P]_\alpha,\varnothing)$
5	$\dfrac{P_1(a)\ \ldots\ P_n(a)\ \ P_1 \triangleq_\alpha \ldots \triangleq_\alpha P_n\ \ I_\alpha(P_1)\ \ldots\ I_\alpha(P_n)}{sat(a,[P_1]_\alpha,n)}\ count(a,[P_1]_\alpha,\{P_1,\ldots,P_n\})$
6	$\dfrac{P_1(a)\ \ldots\ P_n(a)\ \ P_1 \triangleq_\alpha \ldots \triangleq_\alpha P_n\ \ I_\alpha(P_1)\ \ldots\ I_\alpha(P_n)}{count(a,[P_1]_\alpha,S \subset \{P_1,\ldots,P_n\}) \text{ is inapplicable}}\ count(a,[P_1]_\alpha,S)uc$
7	$\dfrac{sat(a,[P]_\alpha,n) \quad sat(b,[P']_\alpha,m) \quad P \triangleq_\alpha P' \quad n > m}{pref_\alpha(a,b)}\ prefinf(a,b,[P]_\alpha)$
8	$\dfrac{sat(a,[Q]_\alpha,n) \quad sat(b,[Q']_\alpha,m) \quad Q \triangleq_\alpha Q' \triangleright_\alpha P \quad n < m}{prefinf(a,b,[P]_\alpha) \text{ is inapplicable}}\ prefinf(a,b,[P]_\alpha)uc$
9	$\dfrac{sat(a,[P]_\alpha,n) \quad sat(b,[P']_\alpha,m) \quad P \triangleq_\alpha P' \quad n = m}{eqpref_\alpha(a,b)}\ eqprefinf(a,b,[P]_\alpha)$
10	$\dfrac{sat(a,[Q]_\alpha,n) \quad sat(b,[Q']_\alpha,m) \quad Q \triangleq_\alpha Q' \quad n \neq m}{eqprefinf(a,b,[P]_\alpha) \text{ is inapplicable}}\ eqprefinf(a,b,[P]_\alpha)uc$

Table 3.4: Inference schemes

such an argument will be defeated by an undercutter built with inference rule 3 if φ is the case. Inference schemes 4 and 5 are used to count the number of interests of equal importance (according to audience α) as some interest P_1 that outcome a satisfies. This type of inference is inspired by accrual [103], 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. Inference scheme 4 can be used when an outcome satisfies no interests. It is possible to construct an argument that does not count all interests that are satisfied, a so-called non-maximal count. But we want all interests to be counted, otherwise we would conclude incorrect preferences. To ensure that only maximal counts are used, we provide an inference scheme to construct arguments that undercut non-maximal counts (inference scheme 6). An argument of this type says that any count which is not maximal is not applicable. Inference scheme 7 says that an outcome a is preferred over an outcome b if the number of interests of a certain importance level that a satisfies is higher than the number of interests on that same level that b satisfies. Inference scheme 8 undercuts scheme 7 if there is a more important level than that of P on which a and b do not satisfy the same number of interests. Finally, inference schemes 9 and 10 do the same as 7 and 8, but for equal preference.

Defeat The most common type of defeat is rebuttal. An argument rebuts another argument if its conclusion contradicts conclusion of the other argument. Conclusions contradict each other if one is the negation of the other, or if they are preference or importance statements that are incompatible (e.g. $\mathit{pref}_\alpha(a,b)$ and $\mathit{pref}_\alpha(b,a)$, or $\mathit{pref}_\alpha(a,b)$ and $\mathit{eqpref}_\alpha(a,b)$). Defeat by rebuttal is mutual. Another type of defeat is undercut. An undercutter is an argument for the inapplicability of an inference used in another argument. 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 3.5. (Defeat) An argument A *defeats* an argument B ($A \to B$) if $\mathtt{conc}(A)$ and $\mathtt{conc}(B)$ are contradictory (*rebuttal*), or $\mathtt{conc}(A)$ =‘$\mathtt{inf}(B)$ is inapplicable’ (*undercut*), or A defeats a subargument of B.

Let us return to the example. With the information from the knowledge base, the arguments A and B in Table 3.5 can be formed. A advocates a preference for c, based on the interest wealth. B advocates a preference for f, based on the interest family. Without an ordering on these interests, no decision between these arguments can be made. But if $\mathit{wealth} \rhd_M \mathit{family}$ is known, argument C can be made, which undercuts B. Similarly, with $\mathit{family} \rhd_M \mathit{wealth}$, argument D can be made, which undercuts A.

Validity If some conditions in the input knowledge base (KB) hold, it can be shown that the proposed argumentation framework models ceteris paribus and lexicographic preference. In the following, we consider a single audience and leave out the subscript α.

Condition 3.1. Let $\mathcal{C}$ be a set of interests to be used as criteria, with importance order $\unrhd$.
(1) For all P, ‘$I(P)$’ is in KB iff $P \in \mathcal{C}$.
(2) For all $P \in \mathcal{C}$, a, ‘$P(a)$’ is a conclusion of a sceptically justified argument iff a satisfies P.
(3) The relative importance among interests is
 (a) a total preorder,
 (b) the identity relation,
and for all $P, Q \in \mathcal{C}$, ‘$P \rhd Q$’ is in KB iff $P \rhd Q$, and ‘$P \triangleq Q$’ is in KB iff $P \triangleq Q$.

$$A:\quad \dfrac{\dfrac{\dfrac{\mathit{highsal}(c) \quad \mathit{highsal}(x) \Rightarrow \mathit{wealth}(x)}{\mathit{wealth}(c)} \quad I_M(\mathit{wealth})}{\mathit{sat}(c, [\mathit{wealth}]_M, 1)} \quad \dfrac{}{\mathit{sat}(f, [\mathit{wealth}]_M, 0)} \quad \mathit{wealth} \triangleq_M \mathit{wealth} \quad 1 > 0}{\mathit{pref}_M(c, f)}\ \alpha$$

$$B:\quad \dfrac{\dfrac{\dfrac{\neg\mathit{full\text{-}time}(f) \quad \neg\mathit{full\text{-}time}(x) \Rightarrow \mathit{family}(x)}{\mathit{family}(f)} \quad I_M(\mathit{family})}{\mathit{sat}(f, [\mathit{family}]_M, 1)} \quad \dfrac{}{\mathit{sat}(c, [\mathit{family}]_M, 0)} \quad \mathit{family} \triangleq_M \mathit{family} \quad 1 > 0}{\mathit{pref}_M(f, c)}\ \beta$$

$$C:\quad \dfrac{\dfrac{\dfrac{\mathit{highsal}(c) \quad \mathit{highsal}(x) \Rightarrow \mathit{wealth}(x)}{\mathit{wealth}(c)} \quad I_M(\mathit{wealth})}{\mathit{sat}(c, [\mathit{wealth}]_M, 1)} \quad \dfrac{}{\mathit{sat}(f, [\mathit{wealth}]_M, 0)} \quad \mathit{wealth} \rhd_M \mathit{family} \quad 1 \neq 0}{\beta \text{ is inapplicable}}$$

$$D:\quad \dfrac{\dfrac{\dfrac{\neg\mathit{full\text{-}time}(f) \quad \neg\mathit{full\text{-}time}(x) \Rightarrow \mathit{family}(x)}{\mathit{family}(f)} \quad I_M(\mathit{family})}{\mathit{sat}(f, [\mathit{family}]_M, 1)} \quad \dfrac{}{\mathit{sat}(c, [\mathit{family}]_M, 0)} \quad \mathit{family} \rhd_M \mathit{wealth} \quad 1 \neq 0}{\alpha \text{ is inapplicable}}$$

Table 3.5: Example arguments

Theorem 3.1. (i) If conditions 3.1.1, 3.1.2 and 3.1.3a hold, then *pref*(a, b) (resp. *eqpref*(a, b)) is a sceptically justified conclusion of the argumentation framework iff a is strictly (resp. equally) preferred over b according to the lexicographic preference ordering.
(ii) If conditions 3.1.1, 3.1.2 and 3.1.3b hold, then *pref*(a, b) (resp. *eqpref*(a, b)) is a sceptically justified conclusion of the argumentation framework iff a is strictly (resp. equally) preferred over b according to the ceteris paribus preference ordering.

Proof. We prove the theorem for strict preference. The same line of argument can be followed for equal preference.
(i) $\Leftarrow$: Suppose a is strictly lexicographically preferred over b. This means that there is an importance level on which a satisfies more interests (say, $P_1, \ldots, P_n$) than b (say, $P'_1, \ldots, P'_m$, $n > m$), and on all more important levels, a and b satisfy an equal number of interests. In this case, we can construct the following arguments, where the first two arguments are subarguments of the third (note that these arguments can also be built if m is equal to 0, by using the empty set count).

$$\frac{P_1(a) \quad \ldots \quad P_n(a) \quad I(P_1) \quad \ldots \quad I(P_n) \quad P_1 \triangleq \ldots \triangleq P_n}{sat(a, [P_1], n)}$$

$$\frac{P'_1(b) \quad \ldots \quad P'_m(b) \quad I(P'_1) \quad \ldots \quad I(P'_m) \quad P'_1 \triangleq \ldots \triangleq P'_m}{sat(b, [P'_1], m)}$$

$$\frac{sat(a, [P_1], n) \quad sat(b, [P'_1], m) \quad P_1 \triangleq P'_1 \quad n > m}{pref(a, b)}$$

We will now try to defeat this argument. Premises of the type $P(a)$ are justified by condition 3.1.2. Premises of the type $I(P)$ and $P_1 \triangleq P_2$ cannot be defeated (conditions 3.1.1 and 3.1.3a). There are three inferences we can try to undercut (the last inference of the argument and the last inferences of two subarguments). For the first count, this can only be done if there is another P_j such that $I(P_j)$ and $P_j \triangleq P$ and $P_j \notin \{P_1, \ldots, P_n\}$ and $P_j(a)$ is the case. However, $P_1 \ldots P_n$ encompass all interests that a satisfies on this level, 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 $P_1 \ldots P_n$ (resp. $P'_1 \ldots P'_m$) into account. Such an undercutter has no defeaters, so any non-maximal count is not justified. The undercutter of *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 $n = m$ for undercutter arguments that use $Q \rhd P$, since we have that on all more important levels than $[P_1]$, a and b satisfy an equal number of interests. So the undercutter inference rule cannot be applied since $n \neq m$ is not true. For that reason, a rebutting argument with conclusion *pref*(b, a) will not be justified. This means that for every possible

type of defeat, either the defeat is inapplicable or the defeater is itself defeated by undefeated arguments. This means that the argument is sceptically justified.
$\Rightarrow$: Suppose that a is not strictly lexicographically preferred over b. This means that for all importance levels $[P]$, either a does not satisfy more interests than b on that level, or there exists a more important level where b satisfies more interests than a. This means that any argument with conclusion $\mathit{pref}(a,b)$ (which has to be of the form above) is either undercut by $\mathit{count}(b,[P],S)\mathit{uc}$ because it uses a non-maximal count, or by $\mathit{prefinf}(a,b,[P])\mathit{uc}$ because there is a more important level where a preference for b over a can be derived. This means that any such argument will not be sceptically justified.
(ii) $\Leftarrow$: Suppose a is strictly ceteris paribus preferred over b. This means that there is (at least) one interest, let us say P, that a satisfies and b does not, and there are no interests that b satisfies and a does not. In this case, we can construct the following argument.

$$\dfrac{\dfrac{P(a) \quad I(P)}{\mathit{sat}(a,[P],1)} \quad \dfrac{}{\mathit{sat}(b,[P],0)} \quad P \triangleq P \quad 1 > 0}{\mathit{pref}(a,b)}$$

Premise $P(a)$ is justified by condition 3.1.2. Premise $I(P)$ cannot be defeated (condition 3.1.1). Note that, since there is no importance ordering specified, counts can only include 0 or 1 interest(s). So the first count cannot be undercut, because there are no other interests that are equally important as P (condition 3.1.3b). The second count cannot be undercut because b does not satisfy P. Since there are no interests that b satisfies but a does not, the last inference can only be undercut by an undercutter that uses a non-maximal count and so will be undercut itself.
$\Rightarrow$: Suppose a is not strictly ceteris paribus preferred over b. This means that either there is no interest that a satisfies but b does not, or there is some interest that b satisfies and a does not. In the first case, the only arguments that derive a preference for a over b have to use non-maximal counts and hence are undercut. In the second case, any argument that derives a preference for a over b is rebut by the following argument,

$$\dfrac{\dfrac{Q(b) \quad I(Q)}{\mathit{sat}(b,[Q],1)} \quad \dfrac{}{\mathit{sat}(a,[Q],0)} \quad Q \triangleq Q \quad 1 > 0}{\mathit{pref}(b,a)}$$

and is not sceptically justified. □

3.7 Conclusion

In this paper we have made a case for explicitly modelling underlying interests when reasoning about preferences in the context of practical reasoning. We have presented an argumentation framework for reasoning about qualitative interest-based preferences that models ceteris paribus and lexicographic preference.

In the current framework, we have only considered Boolean issues and interests. While this suffices to illustrate the main points discussed in this paper, multi-valued scales would be more realistic. Such an approach would open the way to modelling different degrees of (dis)satisfaction of an interest. For example, Amgoud *et al.* [1] take into account the level of satisfaction of goals on a bipolar scale. In the Boolean case, the lexicographic preference ordering is based on counting the number of interests that are satisfied by outcomes. This is no longer possible if multi-valued scales are used. In that case, we could count interests that are satisfied to a certain degree (like e.g. [1]), or compare outcomes in a pairwise fashion and count the number of interests that one outcome satisfies to a higher degree than another (like e.g. [93, 134]).

Currently, we suppose that the interests and importance ordering among them are given in a knowledge base. We can make our framework more flexible by allowing such statements to be derived in a way that is similar to the derivation of statements about the satisfaction of interests.

We would also like to look into the interplay between different issues promoting or demoting the same interest. For example, a high salary and a high position both lead to status, but together they may lead to even more status. Or a low salary may promote cutback, but providing a lease car will demote it. Do these effects cancel each other out? The principles that play a role here are related to the questions posed in the context of accrual of arguments [103].

Since our long-term goal is the development of an automated negotiation support system, we plan to look into negotiation strategies that are based on qualitative, interest-based preferences as described here, as opposed to utility-based approaches currently in use. For the same reason, we plan to implement the argumentation framework for reasoning about interest-based preferences that we have presented here. Another interesting question in this context is how interest-based preferences can be elicited from a human user.

Chapter 4

Qualitative Preference Systems: A framework for qualitative multi-criteria preferences

Abstract A key challenge in the representation of qualitative, multi-criteria preferences is to find a compact and expressive representation. Various frameworks have been introduced, each with its own distinguishing features. In this paper we introduce a new representation framework called Qualitative Preference Systems (QPS), which combines priority and cardinality. Moreover, the framework incorporates knowledge that serves two purposes: to impose (hard) constraints, but also to define new (abstract) concepts. We show that QPSs provide a general, flexible and succinct way to represent conditional preferences, underlying interests, bipolar preferences and preferences based on goals. We compare the framework in detail with two well-known preference representation frameworks. Preferences between outcomes are often derived from orderings on the possible values of variables that may range over a variety of domains. We show that QPSs that are based on such multi-valued criteria can be translated into equivalent goal-based QPSs that are just as succinct, and that goal-based QPSs allow for more fine-grained updates than their multi-valued counterparts. In short, QPS offers a rich and practical representation for qualitative, multi-criteria preferences.

4.1 Introduction

More and more technology is being developed that aims to support people in their activities and decisions. Such systems need an accurate representation of their users' preferences. Human preferences between objects (products, states of the world, agreements etc.) are typically not intrinsic but based on the evaluation of several *criteria*. Also, although preferences are commonly represented by numeric utilities, the research community is looking for representation formats that are close to the *qualitative* way that humans use to reason about preferences. A key challenge in the representation of qualitative, multi-criteria preferences is to find a *compact* and at the same time *expressive* representation. A framework for preference representation provides an adequate tool if it is sufficiently expressive to compactly represent a broad range of preference orderings. To this end, various frameworks have been introduced, each with its own distinguishing features. For example, in lexicographic approaches (e.g. [8]) preference over outcomes is determined by combining multiple criteria according to *priority*. Here, priority is a partial order on criteria. The lexicographic rule induces a preference if every criterion either supports this preference or is 'overruled' by a criterion with higher priority. Some goal-based approaches (e.g. [33, 44]) use *cardinality* and compare alternatives by the number of goals they satisfy.

In this paper we introduce a rich and practical representation framework for qualitative multi-criteria preferences called *Qualitative Preference Systems* (QPS). This framework enables preference representation by using *priority* and *cardinality*. Moreover, the framework incorporates *knowledge* that serves two purposes: as usual, knowledge can be used to impose (hard) *constraints*, but also to define new (abstract) *concepts*. To illustrate, it can represent facts about the world (e.g. Barcelona is in Spain), the feasibility of options (e.g. hotel X is fully booked in July), and definitions (e.g. the cost of a holiday is the sum of the costs of the flight, hotel and food). A fundamental part of the QPS framework is the lexicographic rule studied by Andréka *et al.* [8]. This rule offers a principled tool for *combining basic preferences*. We believe this ability to combine preferences is essential for any practical approach to representing qualitative preferences. It is needed in particular for constructing *multi-criteria* preferences. It is not sufficient, however, since more expressivity is needed to be useful in practice. Therefore, QPSs in addition provide tools for representing knowledge, for abstraction, for counting, and provide a layered structure for representing preference orderings. The QPS framework is presented in detail in Section 4.2.

Section 4.3 illustrates the expressivity of the QPS framework. We first discuss two ways to deal with preferential dependence between attributes: conditional preferences and underlying interests. Both approaches can be modelled in a QPS, though we argue that the second one is more natural and intuitive. Next, we show that QPS provides a general, flexible and succinct way to represent preferences based on goals. In this approach goals are modelled as criteria that can be combined to derive a preference between outcomes. We show that the best-known qualitative approaches to interpret goals as a representation of preference are all expressible in a QPS. We then turn to the case where preferences are bipolar, i.e. based on arguments pro and arguments con. We show that all natural principles to derive preferences from bipolar arguments map straightforwardly to QPS criteria. Finally, we discuss in detail the relation between Qualitative Preference Systems and two well-known frameworks that are representative for a large number of purely qualitative approaches to modelling preferences, namely Logical Preference Description language (LPD) [33] and CP-nets

[29]. We show that LPD can be embedded into the QPS framework and that there is an order preserving embedding of CP-nets in the QPS framework. In addition, we consider the key issue of compact preference representation and show that these embeddings provide a representation that is just as succinct as the LPD expressions and CP-nets.

Section 4.4 introduces a variant of QPS called goal-based Qualitative Preference Systems. Most goal-based approaches in the literature define outcomes as propositional models, i.e. all variables are Boolean, either true or false. In real-world applications however, not all variables are Boolean. For example, variables may be numeric (e.g. cost, length, number, rating, duration, percentage) or nominal (e.g. destination, colour, location). Qualitative Preference Systems typically express preferences, in a compact way, based on preference orderings on the possible values of variables. In Section 4.4 we show that such QPSs can be translated into equivalent goal-based QPSs, i.e. QPSs that express preferences based solely on goals. Such a translation requires at most polynomially more space, and hence is just as succinct as the original QPS. This result shows that goals are very expressive as a representation of qualitative preferences among outcomes.

In Section 4.5 we show that goal-based criterion trees also have some added value compared to trees with multi-valued criteria. We introduce basic updates on a QPS and show that goal-based QPSs allow for more fine-grained updates than their multi-valued counterparts. This is due to the different structure of goal-based criteria. We suggest a top-down approach to preference elicitation that starts with coarse updates and only adapts the criterion structure if more fine-grained updates are needed.

Section 4.6 discusses some related work that was not discussed elsewhere in the paper, and Section 4.7 concludes the paper.

4.2 Qualitative Preference Systems

4.2.1 Elements of a Qualitative Preference System

The main aim of a QPS is to determine preferences between *outcomes* (or *alternatives*) in a purely qualitative way. An outcome is 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*. We distinguish between simple and compound criteria. Simple criteria are based on a single variable. Multiple (simple) criteria can be combined in a compound criterion to determine an overall preference. There are two kinds of compound criteria: cardinality criteria and lexicographic criteria. The subcriteria of a cardinality criterion all have equal priority, and preference is determined by a kind of voting mechanism that counts the number of subcriteria that support a certain preference and those that do not. In a lexicographic criterion, the subcriteria are ordered by priority and preference is determined by the subcriteria with the highest priority; lower priority subcriteria only influence the preference if the higher priority subcriteria are indifferent. The exact definitions of the different kinds of criteria will be given in the next subsections.

Definition 4.1. (Qualitative Preference System) A *Qualitative Preference System (QPS)* is a tuple $\langle Var, Dom, K, \mathcal{C} \rangle$. *Var* is a finite set of *variables*. Every variable $X \in Var$ has a domain $Dom(X)$ of possible values. K (a *knowledge base*) is a set of constraints on the assignments of values to the variables in *Var*. A *constraint* is an equation of the form $X = Expr$ where $X \in Var$ is a variable and *Expr* is an algebraic expression that maps to $Dom(X)$.[1] 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. Ω denotes the set of all outcomes: $\Omega \subseteq \prod_{X \in Var} Dom(X)$. α_X denotes the value of variable X in outcome α. $\mathcal{C}$ is a finite rooted tree of criteria, where leaf nodes are simple criteria and other nodes are compound criteria. Child nodes of a compound criterion are called its subcriteria. The root of the tree is called the top criterion. 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.

Example 4.1. When comparing holidays, some variables could be D (destination), Bb (been there before) and C (cost), with $Dom(D) = \{R, B, P\}$ (Rome, Barcelona and Paris), $Dom(Bb) = \{\top, \bot\}$, $Dom(C) = \mathbb{Z}^+$. The definition of concepts (e.g. the cost of a holiday is the sum of the costs of the flight, hotel and food) can be straightforwardly represented with the following constraint: $C = FlightC + HotelC + FoodC$. Equational constraints are also sufficiently expressive to model different kinds of knowledge. For example, suppose I want to express that I have never been to Barcelona, i.e. in all outcomes where $D = B$, we should have $Bb = \bot$. To do this, we first introduce an auxiliary variable A with $Dom(A) = \{\top, \bot\}$. Then we add $A = (D = B)$ and $A = A \wedge \neg Bb$ to the constraint base K. This ensures that there are no outcomes where $D = B$ and $Bb = \top$: if the first constraint is satisfied, A must be true, but in that case the second constraint is violated.

4.2.2 Simple criteria

A simple criterion specifies a preference ordering on the values of a single variable. Its preference between outcomes is based solely on the value of this variable in the considered outcomes.

Definition 4.2. (Simple criterion) A *simple criterion* c is a tuple $\langle X_c, \trianglerighteq_c \rangle$, where $X_c \in Var$ is a variable, and $\trianglerighteq_c$, a preference relation on the possible values of X_c, is a preorder on $Dom(X_c)$. $\vartriangleright_c$ is the strict subrelation, $\doteq_c$ is the indifference subrelation. A

[1]For now, we represent constraints with algebraic expressions, which are sufficiently expressive for our current purposes. In future we would like to develop a more intuitive language to express constraints and the other components of a QPS.

simple criterion $c = \langle X_c, \trianglerighteq_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $\alpha_{X_c} \trianglerighteq_c \beta_{X_c}$.

Example 4.2. Consider again the holiday domain from Example 4.1. A criterion 'economy' can be defined as $s_1 = \langle C, \trianglerighteq_{s_1} \rangle$ where for all $x, x' \in Dom(C)$, $x \trianglerighteq_{s_1} x'$ iff $x \leq x'$. This criterion prefers any holiday with a lower cost over any holiday with a higher cost, irrespective of the values of other variables. A criterion 'exploration' can be defined as $s_2 = \langle Bb, \{(\perp, \top)\} \rangle$. This criterion prefers holidays with $Bb = \perp$ over holidays with $Bb = \top$.

Note that the preference by a simple criterion is different from the ceteris paribus (all else being equal) approach taken in other frameworks (e.g. [138, 29, 137]). That is, a simple criterion prefers one outcome over another iff the first has a better value for the variable that the criterion is based on than the last, irrespective of the values of other variables.

Observation 4.1. Let $c = \langle X_c, \trianglerighteq_c \rangle$ be a simple criterion. Then $\succeq_c$ is a preorder. If $\trianglerighteq_c$ is total, then so is $\succeq_c$.

In general, the variables of a QPS can have any arbitrary domain and simple criteria can be defined over such variables (for example, the 'economy' criterion in Example 4.2). In the goal-based case however, we define outcomes as propositonal models, and hence all variables are Booleans. Goals are defined as simple criteria on Boolean variables that prefer the truth of a variable over falsehood.

Definition 4.3. (Goal) A QPS *goal* is a simple criterion $\langle X, \trianglerighteq \rangle$, where $X \in Var$ is a Boolean variable ($Dom(X) = \{\top, \perp\}$), and $\top \triangleright \perp$. For convenience, we denote such a goal by its variable X.

This is straightforward when goals are atomic, e.g. p. If goals are complex propositional formulas, e.g. $(p \vee q) \wedge \neg r$, an auxiliary variable s can be defined by the constraint $s = (p \vee q) \wedge \neg r$. As this is a purely technical issue, we will sometimes use the formula instead of the auxiliary variable in order not to complicate the notation unnecessarily.

4.2.3 Cardinality criteria

A cardinality criterion consists of a set of subcriteria, that all have the same priority. It weakly prefers an outcome α over an outcome β if it has at least as many subcriteria that strictly prefer α over β as criteria that do not weakly prefer α over β.

Definition 4.4. (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 $\alpha \succeq_c \beta$, iff $|\{s \in C_c \mid \alpha \succ_s \beta\}| \geq |\{s \in C_c \mid \alpha \not\succeq_s \beta\}|$.

Example 4.3. Consider a cardinality criterion c that has two subcriteria: the 'economy' criterion and the 'exploration' criterion from Example 4.2: $c = \langle\{s_1, s_2\}\rangle$. Consider three outcomes α, β, γ such that $\alpha_D = R$, $\alpha_C = 500$, and $\alpha_{Bb} = \top$; $\beta_D = B$, $\beta_C = 350$, and $\beta_{Bb} = \bot$; and $\gamma_D = P$, $\gamma_C = 700$, and $\gamma_{Bb} = \bot$. Then we have $\beta \succ_c \alpha \approx_c \gamma$. β is the most preferred because it is preferred to α and γ by s_1 and not less preferred by s_2. α and γ are equally preferred because α is more preferred by s_1 and γ is more preferred by s_2.

Unfortunately, transitivity of $\succeq_c$ is not guaranteed for just any set of subcriteria. This problem is known as the Condorcet effect. For example, consider three outcomes α, β, γ and three subcriteria s_1, s_2, s_3 such that $\alpha \succ_{s_1} \beta \succ_{s_1} \gamma$, $\beta \succ_{s_2} \gamma \succ_{s_2} \alpha$, and $\gamma \succ_{s_3} \alpha \succ_{s_3} \beta$. Then α would be strictly preferred over β, β strictly preferred over γ and γ strictly preferred over α, so the preference would not be transitive. However, there are some conditions under which transitivity *can* be guaranteed. E.g. if every subcriterion is a goal $s = \langle X_s, \trianglerighteq_s\rangle$, they all induce a total preorder of preference that stratifies the outcome space into two levels: the outcomes where $X_s = \top$ are more preferred and the outcomes where $X_s = \bot$ are less preferred. This also means that $\alpha \succ_s \beta$ iff $\alpha_{X_s} = \top$ and $\beta_{X_s} = \bot$; and $\alpha \not\succeq_s \beta$ iff $\alpha_{X_s} = \bot$ and $\beta_{X_s} = \top$. So in this case the definition preference by a cardinality criterion compares the number of goals X_s that α satisfies to the number of goals that β satisfies, just as is done by e.g. the # strategy of [33] or the leximin ordering in [44].

Definition 4.5. (Goal-based cardinality criterion) A *goal-based cardinality criterion* is a cardinality criterion all of whose subcriteria are goals.

Proposition 4.1. Let $c = \langle C_c\rangle$ be a goal-based cardinality criterion. Then $\alpha \succeq_c \beta$ iff $|\{s \in C_c \mid \alpha_{X_s} = \top\}| \geq |\{s \in C_c \mid \beta_{X_s} = \top\}|$.

Proof. A goal s is defined as a simple criterion on a Boolean variable X_s (with domain $Dom(X_s) = \{\top, \bot\}$), such that $\top \triangleright_s \bot$. Note that the relation $\trianglerighteq_s$ is total; there are no incomparable values. If we compare two outcomes α and β on such a criterion, we have exactly the following possibilities: if $\alpha_{X_s} = \top$ and $\beta_{X_s} = \top$, or $\alpha_{X_s} = \bot$ and $\beta_{X_s} = \bot$, then $\alpha \approx_s \beta$; if $\alpha_{X_s} = \top$ and $\beta_{X_s} = \bot$, then $\alpha \succ_s \beta$; if $\alpha_{X_s} = \bot$ and $\beta_{X_s} = \top$, then $\beta \succ_s \alpha$. This means that the definition of preference by a goal-based cardinality criterion c is equivalent to $\alpha \succeq_c \beta$ iff $|\{s \in C_c \mid \alpha_{X_s} = \top \& \beta_{X_s} = \bot\}| \geq |\{s \in C_c \mid \alpha_{X_s} = \bot \& \beta_{X_s} = \top\}|$, which is equivalent to $\alpha \succeq_c \beta$ iff $|\{s \in C_c \mid \alpha_{X_s} = \top\}| - |\{s \in C_c \mid \alpha_{X_s} = \beta_{X_s} = \top\}| \geq |\{s \in C_c \mid \beta_{X_s} = \top\}| - |\{s \in C_c \mid \alpha_{X_s} = \beta_{X_s} = \top\}|$, which is equivalent to $\alpha \succeq_c \beta$ iff $|\{s \in C_c \mid \alpha_{X_s} = \top\}| \geq |\{s \in C_c \mid \beta_{X_s} = \top\}|$. □

Proposition 4.2. Let $c = \langle C_c\rangle$ be a goal-based cardinality criterion. Then $\succeq_c$ is a preorder.

Proof. Since all subcriteria of c are reflexive (Observation 4.1), for any outcome α both $|\{s \in C_c \mid \alpha \succ_s \alpha\}|$ and $|\{s \in C_c \mid \alpha \not\succeq_s \alpha\}|$ are 0, so $\alpha \succeq_c \alpha$, hence $\succeq_c$ is reflexive. Since all subcriteria are goals, $\alpha \succeq_c \beta$ iff $|\{s \in C_c \mid \alpha_{X_s} = \top\}| \geq |\{s \in C_c \mid \beta_{X_s} = \top\}|$. This is just a comparison between two integers, and hence is transitive. □

In the following, we will only consider goal-based cardinality criteria, in order to guarantee transitivity of the resulting preference relation.

Andréka *et al.* [8] showed that the only operator to combine *any arbitrary* preference relations that satisfies the desired properties IBUT (independence of irrelevant alternatives, based on preferences only, unanimity with abstentions, and preservation of transitivity) is the priority operator, which assumes that priority is a partial order. We observe here that if only *Boolean* preference relations (such as those resulting from goals) are combined, the cardinality-based rule, in which all combined relations have equal priority, can also be applied as it also satisfies the IBUT properties. Requiring antisymmetry in this case would unneccessarily restrict the expressivity.

4.2.4 Lexicographic criteria

Like a cardinality criterion, a lexicographic criterion combines multiple criteria into one preference ordering. Unlike a cardinality criterion, the set of subcriteria has an associated priority order (a strict partial order, which means that no two subcriteria can have the same priority). A lexicographic criterion weakly prefers outcome α over outcome β if for every subcriterion, either this subcriterion weakly prefers α over β, or there is another subcriterion with a higher priority that strictly prefers α over β. So, if there is a subcriterion that strictly prefers β over α or finds them incomparable, the lexicographic criterion can still prefer α over β, but only if this 'disagreeing' subcriterion is 'overruled' by another subcriterion with higher priority that strictly prefers α over β. If there is no such subcriterion, then the lexicographic criterion will not prefer α over β. This definition of preference by a lexicographic criterion is equivalent to the priority operator as defined by Andréka *et al.* [8]. It generalizes the familiar rule used for alphabetic ordering of words, such that the priority can be any partial order and the combined preference relations can be any preorder.

Definition 4.6. (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 $c = \langle C_c, \rhd_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $\forall s \in C_c(\alpha \succeq_s \beta \vee \exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \rhd_c s))$.

Example 4.4. Consider a lexicographic criterion c that has two subcriteria: the 'economy' criterion and the 'exploration' criterion from Example 4.2, and exploration has higher priority: $c = \langle \{s_1, s_2\}, \{(s_2, s_1)\} \rangle$. For the three outcomes specified in Example 4.3, we have $\beta \succ_c \gamma \succ_c \alpha$. β and γ are preferred to α because they are preferred by s_2, which has the highest priority. β is preferred to γ because they are equally preferred by s_2, but s_1 prefers β. Note that even though α is cheaper than γ and hence preferred by criterion s_1, criterion c prefers γ to α because subcriterion s_2 has higher priority than s_1 and s_2 prefers γ to α.

Proposition 4.3. Let $c = \langle C_c, \rhd_c \rangle$ be a lexicographic criterion. If for all subcriteria $s \in C_c$, $\succeq_s$ is a preorder, then the relation $\succeq_c$ is also a preorder.

Proof. Preservation of reflexivity follows directly from the definition of $\succeq_c$ (if all subcriteria are reflexive, then for every outcome α: $\forall s \in C_c(\alpha \succeq_s \alpha)$ and hence $\alpha \succeq_c \alpha$). Preservation of transitivity has been proven by Andréka *et al.* [8]. □

If there is priority between goals (or if goals have incomparable priority), they can be combined in a goal-based lexicographic criterion. Such a criterion can also be used to specify priority between sets of equally important goals (goal-based cardinality criteria).

Definition 4.7. (Goal-based lexicographic criterion) A *goal-based lexicographic criterion* is a lexicographic criterion all of whose subcriteria are either goals, goal-based cardinality criteria, or goal-based lexicographic criteria.

Note that in the goal-based case, multi-valued simple criteria do not occur anywhere in the criterion tree; that is, all simple criteria are goals. However, we will see later in Section 4.4 that a criterion tree containing multi-valued simple criteria can be translated to an equivalent goal-based criterion tree.

Priority between subcriteria of a lexicographic criterion ($\rhd$) is a strict partial order (a transitive and asymmetric relation). This means that no two subcriteria can have the same priority. If two criteria have the same priority (in which case we assume that they are goals, to avoid Condorcet effects), they have to be combined in a cardinality criterion, which can then be a subcriterion of the lexicographic criterion. To simplify the representation of such a lexicographic criterion with cardinality subcriteria, we define the following alternative specification.

Definition 4.8. (Alternative specification of a lexicographic criterion) A tuple $\langle C'_c, \unrhd'_c \rangle$, where C'_c is a set of criteria and $\unrhd'_c$ is a preorder, specifies a lexicographic criterion $c = \langle C_c, \rhd_c \rangle$ as follows.

- Partition C'_c into priority classes based on $\unrhd'_c$.
- For every priority class P, define a criterion c_P. If P contains only a single criterion s, then $c_P = s$. Otherwise c_P is a cardinality criterion such that for all $s \in P$: $s \in C_{c_P}$.
- Define $c = \langle C_c, \rhd_c \rangle$ such that $C_c = \{c_P \mid P \text{ is a priority class}\}$ and $c_P \rhd_c c_{P'}$ iff for all $s \in P, s' \in P'$: $s \rhd'_s s'$.

For example, the specification $l = \langle \{g_1, g_2, g_3\}, \unrhd \rangle$ such that $g_1 \unrhd g_2 \triangleq g_3$ is short for $l = \langle \{g_1, c\}, \rhd \rangle$ such that $g_1 \rhd c$ and $c = \langle \{g_2, g_3\} \rangle$.

4.3 Expressivity of Qualitative Preference Systems

In this section, we discuss the expressivity of the QPS framework. In Section 4.3.1, we show how conditional preferences and underlying interests can be modelled in a QPS. This is useful when the obvious attributes that define the outcome space are not preferentially independent. Section 4.3.2 discusses how some well-known approaches to derive preferences from goals can be represented in a QPS. Section

4.3.3 discusses bipolar preferences and how they can be modelled in a QPS. Sections 4.3.4 and 4.3.5 compare the QPS framework in detail with two representative approaches to qualitative preference representation: Logical Preference Description language [33] and CP-nets [29].

4.3.1 Conditional preferences and underlying interests

The reason that a QPS uses criteria to define a preference relation between possible outcomes, is that typically, the outcome space is too large for a preference relation to be specified directly. Instead, the outcome space is 'framed' into attributes which may have different values for different outcomes [135]. Simple criteria can then define a preference ordering on those possible values, and then be aggregated to obtain an overall preference between outcomes. The quality of the resulting preference relation depends highly on the chosen framing, i.e. on the attributes that the simple criteria are based on. In some cases, especially in the case of combinatorial or multi-attribute domains, a framing is readily available. For example, in negotiation, outcomes (bids or agreements) are usually defined as assignments of particular values to each of the issues that are on the table. Also in many other situations, objective properties or features of the things to be compared present themselves as attributes on which a preference can be defined.

Although it may be tempting to use a readily available framing to specify preferences over the available outcomes, we first ask the question whether this is actually the best approach. One point of concern is the preferential (in)dependence of the chosen attributes. An attribute X is preferentially independent of an attribute Y if the preference between values of X does not depend on the value of Y. If there are preferential dependencies between attributes, one option is to specify conditional preferences, as is done in e.g. [29, 28]. Below we illustrate how conditional preferences can be modelled in a QPS. Another solution is to use another framing and express the preferences in terms of more fundamental attributes [135, 74]. In recommender systems and decision support, Stolze and Ströbel [115] propose to distinguish between preferences based on the product's features and preferences based on the customer's needs (or intended use of the product). Likewise, Dimopoulos *et al.* [46] distinguish between definitional attributes that establish the set of outcomes, and decisional attributes that are used to identify the best choice among the outcomes. In [126] a case is made for explicitly modelling the reason for the dependencies, and taking the underlying interests of the user as criteria. Usually, there is no one-to-one correspondence between specified attributes and interests, but there are multiple attributes that influence the satisfaction of the same interest. An important question is how to model the interplay between attributes. Below we explore the ways in which interests can be modelled in a QPS.

Conditional preferences A QPS can be used to express conditional preferences, i.e. preferences between values of one variable that depend on the values of other variables.

Example 4.5. If Anne goes on a holiday to Barcelona (*B*), she would like to go together with her friend Juan (*J*), but if she goes to Rome (*R*), she prefers to go with Mario (*M*). To express this conditional preference in a QPS, we use an auxiliary variable *DF*, whose domain consists of all combinations of the variables *D* (destination) and *F* (accompanying friend), i.e. $Dom(DF) = \{(B, J), (B, M), (R, J), (R, M)\}$. To keep the outcomes consistent, the constraint $DF = (D, F)$ is added to the knowledge base. Note that due to this constraint, the addition of the auxiliary variable *DF* does not increase the outcome space. Finally, the following simple criterion expresses the conditional preference: $c = \langle DF, \geq_c \rangle$ where $\geq_c = \{((B, J), (B, M)), ((R, M), (R, J))\}$.

This way of modelling conditional preferences may seem artificial. Instead of representing this kind of preference as conditional preferences on the values of variables, it would be more natural to model the underlying *reason* for the conditional preference, as was argued in [126].

Underlying interests

Example 4.6. Anne likes Juan and Mario equally well and would enjoy a holiday with either of them very much. The reason that she prefers one of them to the other depending on the destination, is that she prefers to go with someone who speaks the language. Thus it would be both more natural and easier to model this preference with a simple criterion based on a Boolean variable *L* (speak the language). The value of this variable would then be defined by a constraint in the knowledge base such as $L = ((D = B \wedge F = J) \vee (D = R \wedge F = M))$. This constraint is still rather ad hoc, and could be replaced by more general knowledge such as 'Barcelona is in Spain', 'the language of Spain is Spanish', 'Juan speaks Spanish', etc.

This example also illustrates the abstraction level for preferences that knowledge provides. It allows one to specify more fundamental goals that are only indirectly related to the most obvious variables with which to specify outcomes. We used only two destinations and two friends to illustrate the principle, but if more options are added, the advantage of modelling underlying reasons or interests becomes even more obvious. In that case, only factual information would have to be added to the knowledge base (e.g. 'Sevilla is in Spain', 'Carmen speaks Spanish'). If the preference is modelled as a conditional preference, for every destination a preference between all friends for company has to be specified separately.

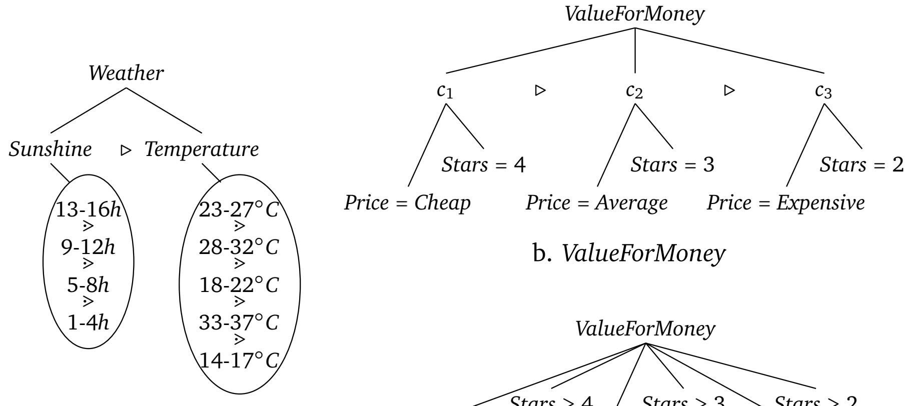
There are different ways to model interests in a QPS. The first option, which was illustrated in Example 4.6, is to model an interest as a simple criterion, based on an 'auxiliary' variable whose values are derived from the values of 'basic' variables. This is the approach taken in [126]. In that paper, all attributes are Boolean, and interests are defined as separate variables, whose truth values depend on the truth values of other attributes. This approach can be generalized to the case where variables have multi-valued domains. There are multiple options. If the variables concerned share the same numeric scale as a domain, we can use the minimum (counterpart of conjunction in the Boolean case) or the maximum (counterpart of disjunction in the

Boolean case). If the domains consist of real numbers, we can also use arithmetic functions such as a (weighted) sum or average. An option that is also applicable if the variable domains are nominal is to specify explicitly what the value of the variable representing the interest should be in different cases.

Example 4.7. Peter and his family are planning a holiday for next summer. Some obvious attributes that can be used to frame the set of possible holidays are the following: the destination, the duration, the total cost, the means of transportation, the kind of accommodation, etc. Peter's children love to swim. Therefore, one of the criteria should be a goal on the Boolean variable *Swimming*. The value of this variable in a given outcome is determined by the constraint (propositional formula) $Swimming = Beach \vee Pool \vee Lake$. Another important issue is the proximity of the accommodation to a supermarket, which Peter defines to be the minimum distance to any supermarket: $ProximityToSM = \min(DistanceToSM1, DistanceToSM2, \ldots)$. An intuitive criterion to compare holiday offers is the price per day. This value can be calculated exactly by dividing the total price by the number of days: $PricePerDay = TotalPrice/NumDays$. Peter's wife likes to see something of the world, and therefore prefers to go somewhere she has not been to before. This can be modelled by a goal $\neg BeenThereBefore$, and a constraint $BeenThereBefore = (Destination = Rome \vee Destination = Prague \vee Destination = Barcelona)$. Another option is a constraint $BeenThereBefore = (Destination = X \wedge beenTo(X))$ and a knowledge base containing formulas like $beenTo(Rome)$, $beenTo(Prague)$, $beenTo(Barcelona)$.

A second possibility to model the relation between interests and other attributes is to model an interest as a compound criterion, whose subcriteria are simple criteria based on 'basic' variables. If the interest is modelled as a lexicographic criterion, the priority between subcriteria can be varied. The interest can also be represented as a cardinality criterion. However, only Boolean simple criteria (goals) can be used as subcriteria of such a cardinality criterion. If the variables that the interest depends on are not Boolean, the simple criteria that are based on them can be translated to equivalent goal-based criteria, as will be explained in Section 4.4.

Example 4.8. For the weather to be considered good (Figure 4.1a), Peter finds that the amount of sunshine is most important. If two climates are equally sunny, then he prefers the one with the best temperature. Figure 4.1b shows a criterion tree where value for money depends on a trade-off between the price and the luxury of an accommodation. In this case the trade-off is modelled with goals of the form $X = v$, where X is a variable and v a possible value from its domain. It is also possible to use a cardinality criterion that counts satisfaction level goals (see Section 4.4.3), as shown in Figure 4.1c. This results in a different preference ordering. While the criterion in Figure 4.1b would prefer an expensive hotel with 4 stars and a cheap hotel with 2 stars to a 3-star hotel with average price, the criterion in Figure 4.1c prefers all three equally.



a. *Weather*

b. *ValueForMoney*

c. *ValueForMoney*

Figure 4.1: Criterion trees for two interests

4.3.2 Goal-based preferences

In planning and decision making, goals are used to identify the desired states or outcomes. Keeney and Raiffa [76] distinguish between objectives and goals: 'An objective generally indicates the 'direction' in which we should strive to do better. A goal is different from an objective in that it is either achieved or not. Goals are useful for clearly identifying a level of achievement to strive toward.' For example, if an objective is to finish a task as soon as possible, a corresponding goal could be to finish before 2 o'clock. If an objective is to find a cheap apartment, a goal could be to find an apartment that costs at most €700 per month. Essentially, goals provide a binary distinction between those states or outcomes that satisfy the goal and those that do not [135]. Outcomes that satisfy all goals are acceptable. However, it may happen that such outcomes are not available, but a decision still has to be made. Or there may be multiple outcomes that satisfy all goals and only one can be chosen. In these situations, goals provide no guidance to choose between the available alternatives [135, 28]. Instead of using goals in an absolute sense, it would be more convenient to use them to derive preferences between outcomes.

Several approaches to derive preferences over outcomes from goals can be found in the literature. Goals are commonly defined as some desired property that is either satisfied or not. As such, it is naturally represented as a propositional formula that can be true or false. Hence outcomes are often defined as propositional models, i.e. valuations over a set of Boolean variables $p, q, r, \ldots$. Sometimes all theoretically possible models are considered, sometimes the set of outcomes is restricted by a set K of knowledge or constraints. In the latter case, it is possible to specify which outcomes are actually available, or to use auxiliary variables whose values are derived from the values of other variables.

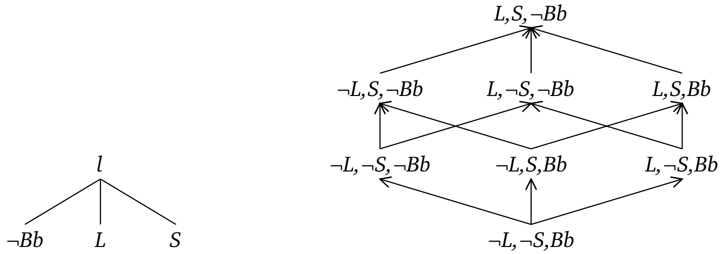
In a QPS, a goal is defined as a simple criterion on a Boolean variable (proposition) that prefers truth over falsehood. Multiple goals can be combined in goal-based compound criteria in order to derive an overall preference. Different interpretations can be found in the literature of what it means, in terms of preferences between outcomes, to have a goal p. In this section, we give a short overview of the best-known ones and show that QPSs can express the same preferences by means of some small examples.

Example 4.9. Anne is planning to go on holiday with a friend. Her overall preference is based on three goals: that someone (she or her friend) speaks the language (L), that it is sunny (S) and that she has not been there before ($\neg Bb$). The set of variables is $Var = \{L, S, Bb\}$. Since every variable is propositional, the domain for each variable is $\{\top, \bot\}$ and there are eight possible outcomes. For the moment we do not constrain the outcome space and do not use auxiliary variables ($K = \varnothing$). Two goals (l and S) are based on atomic propositions, the third ($\neg Bb$) on a propositional formula that contains a negation. The overall preference between outcomes depends on the way that the goals are combined by compound criteria. In the following we discuss several alternatives.

Ceteris paribus preference One interpretation of having a goal p is that p is preferred to $\neg p$ ceteris paribus (all other things being equal) [138, 135, 29]. The main question in this case is what the 'other things' are. Sometimes they are the other variables (atomic propositions) that define the outcomes [29, 138]. Wellman and Doyle [135] define ceteris paribus preferences relative to framings (a factorisation of the outcome space into a cartesian product of attributes). The preference relation over all outcomes is taken to be the transitive closure of the preferences induced by each ceteris paribus preference. So if we have p and q as ceteris paribus goals, then $p \wedge q$ is preferred to $\neg p \wedge \neg q$ since $p \wedge q$ is preferred to $\neg p \wedge q$ (by goal p) and $\neg p \wedge q$ is preferred to $\neg p \wedge \neg q$ (by goal q).

Example 4.10. Consider a lexicographic criterion l that has the three goals as subcriteria, and there is no priority between them, i.e. $l = \langle \{L, S, \neg Bb\}, \varnothing \rangle$ (Figure 4.2a). The resulting preference relation (Figure 4.2b) is a ceteris paribus preference.

This is a general property of Qualitative Preference Systems: a lexicographic criterion with only goals as subcriteria and an empty priority relation induces a ceteris paribus preference, where the other things are defined by the other goals (see also [126]). The main advantage of the ceteris paribus approach is that it deals with multiple goals in a natural, intuitive way. However, the resulting preference relation over outcomes is always partial since there is no way to compare $p \wedge \neg q$ and $\neg p \wedge q$. This is why Wellman and Doyle [135] claim that goals are inadequate as the sole basis for rational action. One way to solve this is to introduce relative importance between goals, which is done in the prioritized goals approach.



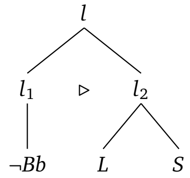
a. Criterion tree b. Preference relation

Figure 4.2: Ceteris paribus preference

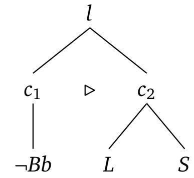
Prioritized goals In e.g. [33, 44], preferences are derived from a set of goals with an associated priority ordering (a total preorder). That is, there are multiple goals, each with an associated rank. There may be multiple goals with the same rank. Various strategies are possible to derive preferences from such prioritized goals. For example, the $\subseteq$ or discrimin strategy prefers one outcome over another if there is a rank where the first satisfies a strict superset of the goals that the second satisfies, and for every more important rank, they satisfy the same goals. The # or leximin strategy prefers one outcome over another if there is a rank where the first satisfies more goals than the second, and for every more important rank, they satisfy the same number of goals.

The prioritized goals strategies discrimin ($\subseteq$) and leximin (#) can also be expressed in a QPS. An exact translation is given in Section 4.3.4. Here we just illustrate the principle. In the prioritized goals approach, priority between goals is a total preorder, which can be expressed by assigning a rank to every goal. A QPS can model a discrimin or leximin preference with a lexicographic criterion that has one subcriterion for every rank. These subcriteria are compound criteria that contain the goals of the corresponding rank, and they are ordered by the same priority as the original ranking. For the discrimin strategy, the subcriteria are lexicographic criteria with no priority ordering between the goals. The leximin strategy uses the number of satisfied goals on each rank to determine overall preference. Therefore, each rank is represented by a cardinality criterion.

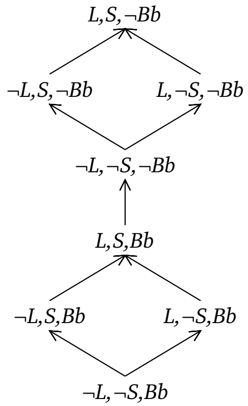
Example 4.11. Suppose that $\neg Bb$ has the highest rank, followed by L and S that have the same rank. The discrimin criterion tree for the example is shown in Figure 4.3a, where l is the top criterion and l_1 and l_2 the lexicographic criteria corresponding to the two ranks. The resulting preference relation is shown in Figure 4.3b. The leximin criterion tree for the example is shown in Figure 4.3c, where l is the top criterion and c_1 and c_2 the cardinality criteria corresponding to the two ranks. The resulting preference relation is shown in Figure 4.3d.



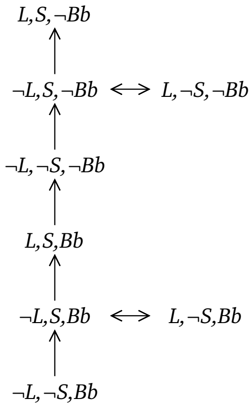
a. Criterion tree



c. Criterion tree



b. Preference relation



d. Preference relation

Figure 4.3: (a, b) Discrimin preference (c, d) Leximin preference

Absolute preference A simple interpretation is that any outcome with p is preferred to any outcome with $\neg p$ [138, 135, 28]. Its use is limited since only one such statement can be made [138]; if we have both $p > \neg p$ and $q > \neg q$, we have that $p \wedge \neg q$ is preferred to $\neg p \wedge q$ and $\neg p \wedge q$ is preferred to $p \wedge \neg q$, which is a contradiction. This means that the interpretation of goals as absolute preferences is not suitable when multiple goals are considered, since the preference statements would be inconsistent.

In a goal-based lexicographic criterion, if there is a single goal that has a strictly higher priority than all other goals, this goal induces an absolute preference. For example, in both Figure 4.3a and Figure 4.3c the goal $\neg Bb$ has higher priority than either of the other goals, and all outcomes satisfying $\neg Bb$ are preferred to all outcomes not satisfying $\neg Bb$, as can be seen in Figure 4.3b and Figure 4.3d. This is a by-effect of the definition of a lexicographic criterion. There can be at most one goal with a strictly higher priority than all other goals, and this correpsonds to the impossibility to have more than one absolute preference.

4.3.3 Bipolar preferences

It has been argued [25, 26, 50, 7] that human preferences are bipolar, i.e. they result from balancing pros and cons. In most approaches that study bipolar preferences, the nature or origin of the arguments pro and arguments con is not specified. One possibility is that they result from satisfaction levels of criteria. This is the case in [133], where value functions are simplified by mapping outcomes to one of three levels of value: satisfactory (1), indifferent (0), or unsatisfactory (-1) (the mapping may differ per person). Two kinds of goals are defined. *Achievement goals* are satisfied if an outcome has a satisfactory value for the attribute that the goal relates to. *Avoidance goals* are satisfied if the outcome does *not* have an unsatisfactory value. Satisfaction of an achievement goal is an argument pro, while non-satisfaction of an avoidance goal is an argument con. The distinction between three levels of satisfaction is quite coarse, but can be useful when one has to find an alternative that is 'good enough' in limited time. The approach can be generalized so that there can be more than one achievement goal and avoidance goal per attribute, as proposed by Amgoud and Prade [7]. It is also possible that there are just two satisfaction levels. In some goal-based approaches (e.g. [33]), it is assumed that not satisfying a goal is something to be avoided, resulting implicitly in both an achievement goal and an avoidance goal for every goal formula that is considered. Another possibility is that arguments pro and arguments con relate to totally different attributes. For example, two-factor theory [62] states that job satisfaction is caused by different factors than job dissatisfaction. For the principles of combining bipolar arguments (and for their representation in a QPS) the origin is not relevant, and we will not make any assumptions about it. The one assumption that we do make is that outcomes are evaluated in the same way: if the fact that outcome α has property p is an argument for outcome α, then if outcome β also has property p, this should also be an argument for outcome β.

In a QPS, it is straightforward to represent achievement and avoidance goals. Here the distinction between achievement goals and avoidance goals is purely conceptual, formally they are both modelled as goals. For example, if it is an achievement goal to have a cost of at most 300 euro, then $C \leq 300$ is a goal criterion, and if it is an avoidance goal not to have a cost of more than 500 euro, then $C < 500$ is a goal criterion.

Bipolar principles Different principles for deriving preferences from bipolar goals are proposed in the literature. Bonnefon *et al.* [25] describe eight heuristics for 'balancing the pros and cons' and report on an experiment to determine which one best predicts the actual choices made by humans. Bonnefon and Fargier [26] and Dubois *et al.* [50] present the same rules, but with different labels. Amgoud and Prade [7] describe several principles to compare decisions based on arguments pro and arguments con. All principles can be categorized along two dimensions. The first dimension is the polarity [7]. Unipolar rules use only achievement goals or only avoidance goals to determine preference. Bipolar rules combine both kinds of goals, but do not compensate between them. Finally, non-polar rules are based on an aggregation of all goals, where compensation is possible. The second dimension relates to the way that the satisfaction of achievement goals and/or avoidance goals by outcomes determines preference between those outcomes. Three different approaches can be found in the literature: counting the number of satisfied goals, comparing the order of magnitude (the maximal importance) of satisfied goals, and a combination of counting and importance in a lexicographic way. The counting approach does not take the relative importance of the goals into account. The other two approaches do, where it is assumed that all goals are ordered by importance by a total preorder.

In the following, we will describe the various principles and show how they can be modelled in a QPS. To illustrate, we assume that there are six achievement goals $G_1^+, \dots, G_6^+$ and six avoidance goals $G_1^-, \dots, G_6^-$, such that $G_1^+ \triangleq G_2^+ \triangleq G_1^- \triangleq G_2^- \triangleright G_3^+ \triangleq G_4^+ \triangleq G_3^- \triangleq G_4^- \triangleright G_5^+ \triangleq G_6^+ \triangleq G_5^- \triangleq G_6^-$. Note that both an achievement goal G^+ and an avoidance goal G^- are goals in the sense that satisfaction (truth of the goal proposition) is preferred over non-satisfaction (falsehood of the goal proposition). The distinction is just that G^+ signifies some satisfactory level, while $\neg G^-$ signifies some unsatisfactory level.

Counting A simple approach is to count the number of goals that are satisfied, without considering relative importance. The unipolar pro variant (Definition 6 in [7]) weakly prefers outcome α to outcome β if α satisfies at least as many achievement goals as β. The unipolar con variant (Definition 9 in [7]) weakly prefers outcome α to outcome β if α satisfies at least as many avoidance goals as β. The bipolar variant (Definition 12 in [7]) combines these two notions and weakly prefers outcome α to outcome β if α satisfies at least as many achievement goals as β and α satisfies at least as many avoidance goals as β. This preference relation is not complete; for example if α satisfies more achievement goals but less avoidance goals than β, they are incomparable. Finally, the non-polar variant (Definition 16 in [7], C1 in [25]) really aggregates all goals and weakly prefers outcome α to outcome β if α satisfies at least as many (achievement or avoidance) goals as β. In this last case, satisfaction of an achievement goal can compensate for non-satisfaction of an avoidance goal and vice versa.

As the unipolar variants of this approach just compare the number of achievement goals resp. avoidance goals that are satisfied by the outcomes that are compared, this can straightforwardly be modelled in a QPS with a cardinality criterion. Figure 4.4a and 4.4b show the criterion trees that model these variants. In the bipolar case, the criteria c_{AchG} and c_{AvG} are combined in a lexicographic criterion as shown in Figure 4.4c. To get the intended result, there should be no priority between c_{AchG} and c_{AvG}. Finally, the non-polar variant, which counts all satisfied achievement and avoidance goals together, is displayed in Figure 4.4d.

Order of magnitude The order of magnitude approach compares the maximal importance of the goals that are (not) satisfied by the outcomes. The unipolar pro

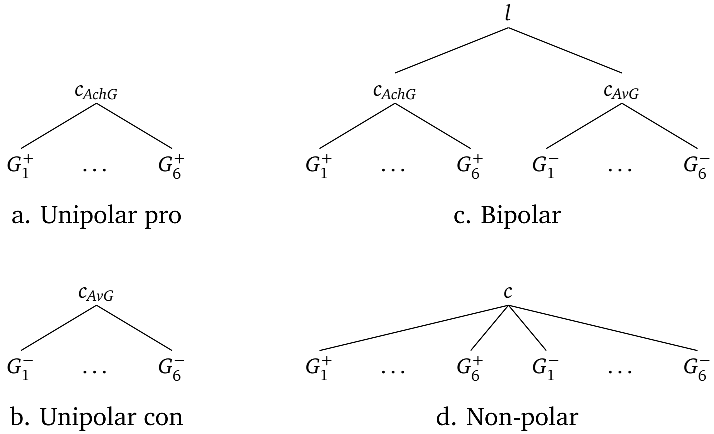


Figure 4.4: Bipolar preferences in a QPS (counting)

variant (Definition 7, 'promotion focus principle' in [7]) weakly prefers outcome α to outcome β if the most important achievement goal that α satisfies is at least as important as the most important achievement goal that β satisfies. The unipolar con variant (Definition 10, 'prevention focus principle' in [7]) weakly prefers outcome α to outcome β if the most important avoidance goal that β does not satisfy is at least as important as the most important avoidance goal that α does not satisfy. The bipolar variant (Definition 13 in [7], F3 in [25]) again combines these two notions and weakly prefers outcome α to outcome β if the most important achievement goal that α satisfies is at least as important as the most important achievement goal that β satisfies and the most important avoidance goal that β does not satisfy is at least as important as the most important avoidance goal that α does not satisfy. Some non-polar order of magnitude rules are also defined (F1, I1, F2, I2 in [25]), but as the resulting preference relations are not transitive, we will not discuss them here. In general, the rules of the order of magnitude approach suffer from a so-called drowning effect, and are not unanimous with abstentions if applied to multi-criteria preferences. For example, if outcome α satisfies one achievement goal of the highest importance and no other goals, and outcome β satisfies many more achievement goals of the highest importance, then α and β would be preferred equally in this approach, while one might intuitively prefer β.

Although the order of magnitude principle does not translate very naturally to a QPS criterion, such a translation is possible. The unipolar pro principle can be modelled by a lexicographic criterion with a goal subcriterion for every importance level, such that the goal for level l is a disjunction of all achievement goals with an importance of at least l (see Figure 4.5a). The unipolar con principle is modelled in a similar way, but the goal for level l is a conjunction of all avoidance goals with an importance of at least l (see Figure 4.5b). This representation is inspired by Brewka's

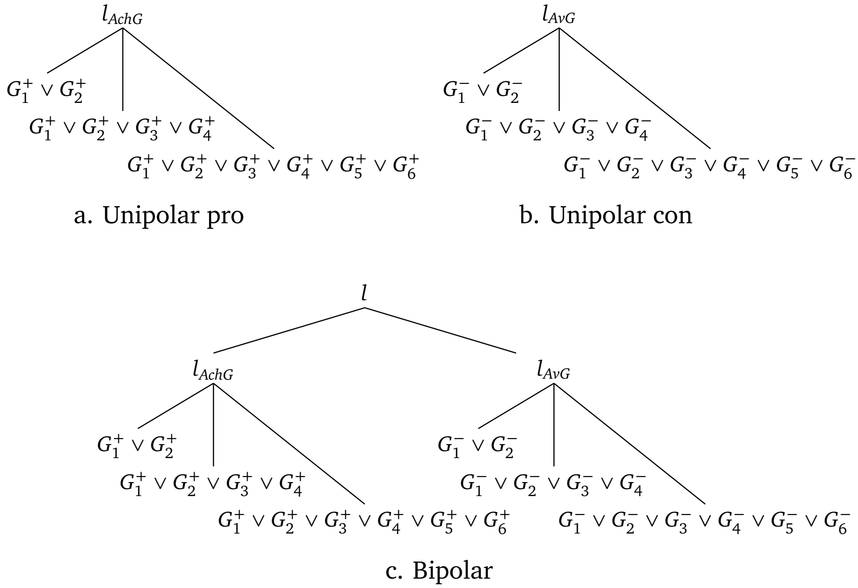


Figure 4.5: Bipolar preferences in a QPS (order of magnitude)

[33] transformation of the $\top$ and κ strategies (which are equivalent to the unipolar pro and con rules) to the # or $\subseteq$ strategies, which map more naturally to a QPS criterion. The bipolar variant of the order of magnitude principle is again modelled by a lexicographic criterion that combines the two unipolar variants without priority, as shown in Figure 4.5c.

Lexi The last approach, lexi, also takes the relative importance of the goals into account, but does not suffer from the drowning effect. The unipolar pro variant (Definition 8 in [7]) weakly prefers outcome α to outcome β if on every importance level, either α satisfies at least as many achievement goals as β, or there is a more important level where α satisfies more achievement goals than β. The unipolar con variant (Definition 11 in [7]) weakly prefers outcome α to outcome β if on every importance level, either α satisfies at least as many avoidance goals as β, or there is a more important level where α satisfies more avoidance goals than β. The bipolar variant (unnumbered in [7], C2 in [25]) combines these two notions and weakly prefers outcome α to outcome β if on every importance level, either α satisfies at least as many achievement goals as β, or there is a more important level where α satisfies more achievement goals than β, and on every importance level, either α satisfies at least as many avoidance goals as β, or there is a more important level where α satisfies more avoidance goals than β. Finally, the non-polar variant (C3 in [25]) weakly prefers outcome α to outcome β if on every importance level, either α

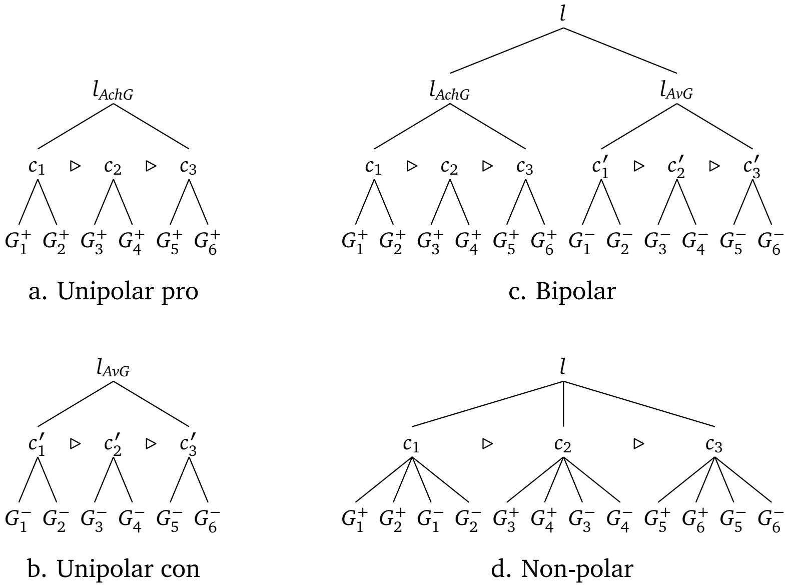


Figure 4.6: Bipolar preferences in a QPS (lexi)

satisfies at least as many (achievement or avoidance) goals as β, or there is a more important level where α satisfies more (achievement or avoidance) goals than β. Note that the counting approach can be seen as a special case of the lexi approach, where all arguments (are assumed to) have the same importance.

The lexi principle can be modelled very naturally in a QPS. For the unipolar pro variant (see Figure 4.6a), a lexicographic criterion is defined with for every importance level a cardinality subcriterion that counts the number of satisfied achievement goals on that level. The subcriteria are prioritized according to importance. The representation of the unpiolar con variant (see Figure 4.6b) is the same except that it considers avoidance goals instead of achievement goals. The bipolar variant combines the two unipolar variants without priority (see Figure 4.6c). Finally, in the non-polar variant, for every importance level, there is a cardinality criterion that counts both the satisfied achievement goals and the satisfied avoidance goals (see Figure 4.6d).

The distinction between negative and positive preferences is also made in the field of soft constraint satisfaction. In [19], negative preferences are taken to be constraints that specify impossible solutions, while positive preferences can be used to discriminate between possible solutions. This can be modelled in a QPS by taking any representation of a bipolar variant (Figure 4.4c, 4.5c or 4.6c) and adding priority: $c_{AvG} \rhd c_{AchG}$ or $l_{AvG} \rhd l_{AchG}$. This makes sure that all possible solutions are preferred to impossible ones, and that the possible solutions are further ordered by the achievement goals that they satisfy. Note however that in a QPS, really impossible outcomes can also be filtered out by constraints in the knowledge base, which may be a more accurate solution.

We have shown that QPSs are suitable to represent bipolar preferences in a straightforward way. On the other hand, one may question the cognitive plausibility of bipolar preferences themselves, seeing that Bonnefon *et al.* [25] found that out of the eight proposed principles, the non-polar lexi rule has the highest empirical validity, i.e. best predicts the actual choices made by humans in their experiments. This rule does not distinguish between achievement goals and avoidance goals and hence is not truly bipolar.

4.3.4 Comparison with Logical Preference Description language

Brewka [33] presents a rank-based description language for qualitative preferences called *logical preference description* language (LPD). The basic expressions of LPD are called *basic preference descriptions* which are pairs $\langle s, R\rangle$ with s one of the *strategy identifiers* $\top$, κ, $\subseteq$, $\#$ and R a *ranked knowledge base* (RKB). An RKB is a set F of propositional formulas together with a *total* preorder $\geq$ on F, representing the relative importance of the formulas. Alternatively, an RKB can be represented as a set of ranked formulas (f, i) where f is a propositional formula and i, the rank of f, is a non-negative integer such that $f_1 \geq f_2$ iff $rank(f_1) \geq rank(f_2)$.

The four basic strategy identifiers refer to different strategies to obtain preferences over outcomes from an RKB. Outcomes in this context are truth assignments or propositional models, i.e. the variables used are Boolean. $\subseteq$ prefers α over β if there is a rank where α satisfies a superset of the formulas that β satisfies, and α and β satisfy the same more important formulas. $\#$ prefers α over β if there is a rank where α satisfies more formulas than β, and for all more important ranks, α and β satisfy the same number of formulas. Since Brewka shows that basic preference descriptions $\langle\top, R\rangle$ and $\langle\kappa, R\rangle$ can be transformed into equivalent basic preference descriptions of the form $\langle\subseteq, R'\rangle$, we do not discuss these strategies here. We note that Coste-Marquis *et al.* [44] also discuss the strategies κ, $\subseteq$ and $\#$ but use different labels.

Definition 4.9. (Translation of an LPD basic preference description to a QPS) A basic preference description $d = \langle s, R\rangle$ can be translated into a QPS $\tau(d) = \langle Var, Dom, K, C\rangle$. Let $R = \langle F, \geq\rangle$ be an RKB. To construct the corresponding QPS from the RKB, the propositional variables used in F are collected in *Var*; moreover, for each formula $f \in F$ a new variable X_f is added to *Var* and $X_f = f$ is added to the knowledge base K (it is not necessary to add constraints $X_f = f$ for *atomic* formulas f but it is convenient below to do so anyway). Clearly, $Dom(X) = \{\top, \bot\}$ for all $X \in Var$. For every formula $f \in F$, a goal on the associated variable is defined: $c_f = \langle X_f, \{(\top, \bot)\}\rangle$. If $s =\subseteq$, preference of $\langle s, R\rangle$ is captured by a lexicographic criterion $c = \langle C_c, \rhd_c\rangle$ such that $C_c = \{c_f \mid f \in F\}$ and $c_f \rhd c_{f'}$ iff $f > f'$. If $s = \#$, for every rank i in R, a cardinality criterion is defined with as subcriteria all goals associated to a formula of that rank: $c_i = \langle\{c_f \mid (f,i) \in R\}\rangle$. The preference of $\langle s,R\rangle$ is captured by a lexicographic criterion $c = \langle C_c, \rhd_c\rangle$ such that $C_c = \{c_i \mid (f',i) \in R\}$ and $c_i \rhd c_{i'}$ iff $i > i'$. This way, a subcriterion of c corresponds with a rank in the RKB R.

Theorem 4.1. Let $d = \langle s,R\rangle$, where $s = \#$ or $s =\subseteq$, be a basic preference description and $\tau(d) = \langle Var, Dom, K, \mathcal{C}\rangle$ its translation to a QPS with top criterion c. Then $\alpha \geq_s^R \beta$ iff $\alpha \geq_c \beta$ for arbitrary outcomes α, β.

Proof. If $s =\subseteq$, goals that correspond to formulas with the same rank are incomparable according to the criterion c. This ensures that an outcome α can only be preferred to an outcome β on some rank, if there is no criterion that strictly prefers β over α, i.e. there is no formula that β satisfies but α does not. This means that α satisfies a superset of the formulas that β satisfies, which is the definition of preference by the $\subseteq$ strategy. If $s = \#$, the way in which preferences are induced by c and its subcriteria corresponds with how the strategy # induces preferences over outcomes. □

Corollary 4.1. The QPS corresponding to a basic preference description is *just as succinct* as this description. That is, the size of the QPS is comparable to that of the LPD description (the size differs at most by a constant factor).

In LPD, *complex preference descriptions* can be built from basic ones with the connectives $\wedge$, $\vee$, $>$ and $-$. The meaning of a complex description is defined in terms of the orderings $\geq_1$ and $\geq_2$ induced by basic preference descriptions d_1 and d_2. The order denoted by $d_1 \wedge d_2$ is the intersection $\geq_1 \cap \geq_2$ (Pareto ordering), $d_1 \vee d_2$ denotes the *transitive closure* of $\geq_1 \cup \geq_2$, $-d_1$ denotes the reversed ordering $\geq_1$, and $d_1 > d_2$ denotes the lexicographic ordering of $\geq_1$ and $\geq_2$ where α is strictly preferred to β if $\alpha >_1 \beta$ or $\alpha \geq_1 \beta$ and $\alpha >_2 \beta$.

We show that complex descriptions can also be translated into a QPS that is just as succinct. To this end, we first introduce the notion of a reversed criterion that induces the reverse of the ordering induced by the original criterion. This can be achieved by reversing the value preferences of all the simple criteria in a QPS.

Definition 4.10. (Reverse of a criterion)

- The *reverse of a simple criterion* $c = \langle X_c, \geq_c\rangle$ is $c^- = \langle X_c, \geq_{c^-}\rangle$ with $\beta \geq_{c^-} \alpha$ iff $\alpha \geq_c \beta$.
- The *reverse of a cardinality criterion* $c = \langle C_c\rangle$ is $c^- = \langle C_{c^-}\rangle$ where $C_{c^-} = \{s_i^- \mid s_i \in C_c\}$.
- The *reverse of a lexicographic criterion* $c = \langle C_c, \rhd_c\rangle$ is $c^- = \langle C_{c^-}, \rhd_{c^-}\rangle$ where $C_{c^-} = \{s_i^- \mid s_i \in C_c\}$ and $s_1^- \rhd_{c^-} s_2^-$ whenever $s_1 \rhd_c s_2$.

Theorem 4.2. Let c_1 and c_2 be any two criteria.

1. The lexicographic criterion $c_{1\wedge 2} = \langle\{c_1, c_2\}, \varnothing\rangle$ induces the order $\geq_{c_1} \cap \geq_{c_2}$.
2. The lexicographic criterion $c_{1>2} = \langle\{c_1, c_2\}, \{(c_1, c_2)\}\rangle$ induces the order $\alpha \geq_{c_{1>2}} \beta$ iff $\alpha >_{c_1} \beta$ or $\alpha \geq_{c_1} \beta$ and $\alpha >_{c_2} \beta$.
3. The criterion c_1^- induces the order $\beta \geq_{c_1^-} \alpha$ iff $\alpha \geq_{c_1} \beta$.

Theorem 4.2 clearly shows the expressive power of QPSs. It is very easy to represent specific operations for combining preference orderings by means of QPSs such as creating a Pareto order ($\wedge$ operator), refining a preference ordering by means of a second one ($>$ operator), and reversing an ordering ($-$ operator). Moreover, Theorem 4.2 shows this can be done just as succinctly with QPSs as with RKBs; i.e. the size needed differs at most with a constant factor.

The only operator that cannot be represented in a QPS is disjunction ($\vee$). However, it has been argued convincingly by Andréka *et al.* [8] that this is not a natural operator, since it does not satisfy the desired properties 'indifference to irrelevant alternatives' and 'unanimity with abstentions'. Indifference to irrelevant alternatives means that two outcomes can be compared solely on their own merits; the presence or absence of other possible outcomes does not influence the preference. The disjunction operator is not indifferent to irrelevant alternatives since it considers the transitive closure of the union of preference relations. Unanimity with abstentions means that if all combined preference relations prefer outcome α over outcome β, except possibly some that are indifferent, then the overall preference relation also prefers α over β. The disjunction operator would be indifferent as soon as one of the combined relations is indifferent, even if all others strictly prefer α over β.

We have shown that LPD descriptions (except disjunction) can be represented by QPSs just as succinctly. QPSs are more general, however, than LPD, which is based on ranked knowledge bases. Whereas RKBs require a total preorder on formulas, QPSs allow incomparable priority between subcriteria. QPSs are not restricted to Boolean variables as LPD is. Apart from propositional formulas, QPSs support the use of equational constraints over arbitrary domains. In particular, QPSs provide a definitional mechanism in order to introduce new concepts (abstract variables) and it is possible to define preferences over such abstract variables. The knowledge that can be captured in a QPS therefore is more general.

4.3.5 Comparison with CP-nets

Boutilier *et al.* [29] introduce CP-nets: qualitative graphical representations of preferences that reflect (conditional) preference statements under a *ceteris paribus* (all else being equal) interpretation.

Definition 4.11. (CP-net [29]) A *CP-net* N over variables $\mathbf{V} = \{X_1, \ldots, X_n\}$ is a directed graph G over $X_1, \ldots, X_n$ whose nodes are annotated with conditional preference tables $CPT(X_i)$ for each $X_i \in \mathbf{V}$. Each conditional preference table $CPT(X_i)$ associates a total order $\succeq^i_{\mathbf{u}}$ with each instantiation $\mathbf{u}$ of X_i's parents $Pa(X_i) = \mathbf{U}$.

A preference ranking $\succeq$ (a total preorder over the set of outcomes) *satisfies* a CP-net N iff for each variable X_i and for each assignment $\mathbf{u}$ to the variables in $\mathbf{U}$, $\mathbf{y}x\mathbf{u} \succeq \mathbf{y}x'\mathbf{u}$ whenever $x \succeq^i_{\mathbf{u}} x'$ – for all assignments $\mathbf{y}$ to the set of variables $\mathbf{Y} = \mathbf{V} - (\mathbf{U} \cup \{X_i\})$ and all $x, x' \in Dom(X_i)$.

N entails $\alpha \succ \beta$, written $N \vDash \alpha \succ \beta$, iff $\alpha \succ \beta$ holds in every preference ordering that satisfies N. Boutilier *et al.* [29] show that $N \vDash \alpha \succ \beta$ iff there is a sequence

of improving flips from β to α. An improving flip of outcome $\mathbf{u}x\mathbf{y}$ with respect to variable X_i is any outcome $\mathbf{u}x'\mathbf{y}$ such that $x' \succ^i_{\mathbf{u}} x$.

We restrict ourselves to acyclic CP-nets here, since cyclic CP-nets may not be satisfiable.

Definition 4.12. (Translation of an acyclic CP-net to a QPS) An acyclic CP-net N over variables $\mathbf{V} = \{X_1, \ldots, X_n\}$ can be translated into a QPS $\tau(N) = \langle Var, Dom, K, \mathcal{C}\rangle$ with a criterion $c \in \mathcal{C}$ as follows. All variables in the CP-net are also variables in the QPS: $\mathbf{V} \subseteq Var$. For every variable $X_i \in \mathbf{V}$, a simple criterion c_i is specified. If X_i is conditionally independent, $c_i = \langle X_i, \succeq_{c_i}\rangle$ such that $x \succeq_{c_i} x'$ iff $x \succ^i x'$. If X_i is preferentially dependent, an auxiliary variable X_i' is added to Var such that $Dom(X_i') = \prod\{Dom(X) \mid X \in X_i \cup Pa(X_i)\}$; the constraint $X_i' = \prod(X_i \cup Pa(X_i))$ is added to K (this procedure was illustrated in Example 4.5); and $c_i = \langle X_i', \succeq_{c_i}\rangle$ such that $x\mathbf{u} \succeq_{c_i} x'\mathbf{u}$ iff $x \succ^i_{\mathbf{u}} x'$. Finally, the top criterion of $\tau(N)$ is a lexicographic criterion $c = \langle C_c, \rhd_c\rangle$ such that for every simple criterion c_i thus generated from the CP-net, $c_i \in C_c$, and $\rhd_c$ is the transitive closure of $\rhd_c'$, where $c_i \rhd_c' c_j$ iff $X_i \in Pa(X_j)$ (note that since N is acyclic, $\rhd_c$ is asymmetric).

Theorem 4.3. Let N be an acyclic CP-net N over variables $\mathbf{V} = \{X_1, \ldots, X_n\}$ and $\tau(N) = \langle Var, Dom, K, \mathcal{C}\rangle$ its translation to a QPS with top criterion c. Then if $N \models \alpha \succ \beta$ then $\alpha \succ_c \beta$ for arbitrary outcomes α, β.

Proof. Suppose that $N \models \alpha \succ \beta$. This means that there is a sequence of improving flips from β to α. First consider the case where this sequence has length 1, i.e. there is a single improving flip w.r.t. some variable X_i from β to α. Since the preference by a simple criterion is taken from the corresponding CPT, $\alpha \succ_{c_i} \beta$. If X_i is not a parent of any variable, all other simple criteria are indifferent between α and β (since they do not involve X_i), so $\alpha \succ_c \beta$. If X_i is a parent of another variable X_j, flipping its value influences the value of the auxiliary variable X_j'. However, c_i has higher priority than c_j, so again we have $\alpha \succ_c \beta$. Since $\succeq_c$ is transitive, we also have $\alpha \succ_c \beta$ if the sequence of improving flips from β to α is longer than 1. □

Corollary 4.2. $\succeq_c$ is an order preserving approximation of the preference relation induced by N, and also satisfies the ceteris paribus property [48].

Corollary 4.3. The QPS corresponding to an acyclic CP-net is *just as succinct* as this description. That is, the size of the QPS is comparable to that of the CP-net (the size differs at most by a constant factor).

There are some things that CP-nets cannot express, but a QPS can. Most importantly, we are able to express abstract preferences based on auxiliary variables whose values are constrained by the knowledge base. Consider the well-known example from game theory called the 'battle of the sexes': a husband and wife have to decide whether to go to the theater or to a football match. The wife prefers the theater and the husband prefers football, but both would rather go together than go to different places. If we let A (resp. B) stand for 'the wife (resp. the husband) goes to the theater' and $\neg A$ (resp. $\neg B$) for 'the wife (resp. the husband) goes to the football match', then the ordering $AB > \neg A \neg B > A \neg B > \neg AB$ represents the wife's preferences. A CP-net cannot express this ordering, since there is no improving flip between $\neg A \neg B$ and AB. In a QPS, this preference can be easily expressed by introducing an auxiliary variable T ('together'), whose values are constrained by $T = A \leftrightarrow B$. A lexicographic criterion with two goals as subcriteria, based on T and A respectively, where the one based on T has higher priority, induces the desired preference ordering $TAB > T \neg A \neg B > \neg TA \neg B > \neg T \neg AB$.

Second, we add priority between criteria, which allows us to express that a good value for one variable is more important than a good value for another variable. TCP-nets [30] are an extension to CP-nets in which some priority between variables is taken into account, but this is not strong enough to represent lexicographic preferences [136]. Wilson's approach [136] can handle such preferences, but does not allow to use auxiliary variables and knowledge as described above.

Third, in the formal definition of CP-nets, preference between values of a variable as given in a CPT, are required to be a total order. According to Boutilier *et al.* [29], a total preorder is allowed, but 'one must be careful not to express indifference between two values of a variable, yet express a (strict) conditional preference for a child of that variable that depends on the values for which the user is indifferent', otherwise the CP-net is not satisfiable. QPSs do not need such restrictions; simple criteria allow any (possibly partial) preorder for the preference between values of a variable.

Fourth, in a CP-net, every variable occurs exactly once. In a QPS, some variables may not occur in any criterion, and some variables may occur in multiple criteria, e.g. if the preference on its values is different from different perspectives, or if the preferences of multiple people are combined.

4.4 Goal-based Qualitative Preference Systems

Preferences in a QPS are ultimately based on simple criteria, i.e. preferences over the values of a single variable. In general, the domain of such a variable may consist of many possible values. In the goal-based case, simple criteria are based on binary goals. In this section we show that the goal-based case is very expressive, by showing that every QPS can be translated into an equivalent goal-based QPS (provided that the domains of the variables used in the original QPS are finite). Moreover, we show that this translation is just as succinct as the original representation. For illustration purposes we will use the following example QPS.

Example 4.12. As before, the outcomes that we compare are holidays. For illustration purposes we simplify the outcome space and consider just two variables: C (cost) and D (destination), where $Dom(C) = \{300, 400, 500\}$ and $Dom(D) = \{R, B, P\}$ (Rome, Barcelona and Paris). For the moment, we do not use any constraints. Preferences are determined by a lexicographic criterion l with two simple subcriteria:

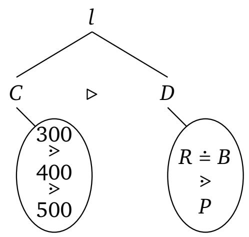


Figure 4.7: Example criterion tree

$\langle C, \geq_C \rangle$ such that $300 >_C 400 >_C 500$ and $\langle D, \geq_D \rangle$ such that $R \doteq_D B >_D P$. We slightly abuse notation and refer to these criteria by their variable, i.e. C and D. C has higher priority than D: $C \rhd_l D$. The criterion tree is shown in Figure 4.7.

We will show how QPSs such as the one in the example can be translated to an equivalent goal-based QPS. In order to do this, we must first formalize the concept of equivalence between QPSs.

4.4.1 Equivalence

An obvious interpretation of equivalence between criteria is the equivalence of the preference relations they induce. I.e. two criteria c_1 and c_2 are equivalent if for all outcomes α, β, we have $\alpha \succeq_{c_1} \beta$ iff $\alpha \succeq_{c_2} \beta$. However, this definition only works if the criteria are defined with respect to the same outcome space, i.e. the same set of variables *Var*, the same domains *Dom* and the same constraints K. Since we will make use of auxiliary variables, we cannot use this definition directly. Fortunately, this is a technical issue that can be solved in a straightforward way.

Definition 4.13. (Equivalence of outcomes) Let $S_1 = \langle Var_1, Dom_1, K_1, \mathcal{C}_1 \rangle$ and $S_2 = \langle Var_2, Dom_2, K_2, \mathcal{C}_2 \rangle$ be two QPSs such that $Var_1 \subseteq Var_2$, $\forall X \in Var_1(Dom_1(X) \subseteq Dom_2(X))$ and $K_1 \subseteq K_2$. Let Ω_1 and Ω_2 denote the outcome spaces of S_1 and S_2, respectively. Two outcomes $\alpha \in \Omega_1$ and $\beta \in \Omega_2$ are *equivalent*, denoted $\alpha \equiv \beta$, iff $\forall X \in Var_1 : \alpha_X = \beta_X$.

In the following, the only variables that are added are auxiliary variables. Such variables do not increase the outcome space because their value is uniquely determined by the values of (some of) the existing variables. We use special variable names of the form '$X = v$' to denote a Boolean variable that is true if and only if the value of variable X is v. For example, the variable $C = 300$ is true in the outcomes where the cost is 300, and false in the other outcomes. When only auxiliary variables are added, every outcome in Ω_1 has exactly one equivalent outcome in Ω_2. We will represent such equivalent outcomes with the same identifier.

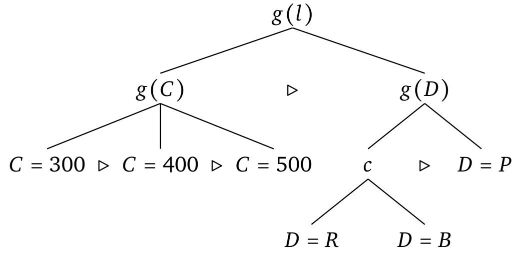


Figure 4.8: Goal-based translation of the criterion tree in Figure 4.7

Definition 4.14. (Equivalence of criteria) Let $S_1 = \langle Var_1, Dom_1, K_1, \mathcal{C}_1 \rangle$ and $S_2 = \langle Var_2, Dom_2, K_2, \mathcal{C}_2 \rangle$ be two QPSs such that $Var_1 \subseteq Var_2$, $\forall X \in Var_1(Dom_1(X) \subseteq Dom_2(X))$ and $K_1 \subseteq K_2$. Let Ω_1 and Ω_2 denote the outcome spaces of S_1 and S_2, respectively. Two criteria c in $\mathcal{C}_1$ and c' in $\mathcal{C}_2$ are called *equivalent* iff $\forall \alpha, \beta \in \Omega_1, \forall \alpha', \beta' \in \Omega_2$, if $\alpha \equiv \alpha'$ and $\beta \equiv \beta'$, then $\alpha \succeq_c \beta$ iff $\alpha' \succeq_{c'} \beta'$.

Definition 4.15. (Equivalence of QPSs) Let $S_1 = \langle Var_1, Dom_1, K_1, \mathcal{C}_1 \rangle$ and $S_2 = \langle Var_2, Dom_2, K_2, \mathcal{C}_2 \rangle$ be two QPSs. S_1 and S_2 are *equivalent* iff the top criterion of $\mathcal{C}_1$ is equivalent to the top criterion of S_2.

4.4.2 From multi-valued criteria to goals

A simple criterion on a variable with a finite domain can be translated to an equivalent goal-based criterion in the following way.

Definition 4.16. (Goal-based translation) Let $c = \langle X, \trianglerighteq \rangle$ be a simple criterion such that $Dom(X)$ is finite. The translation of c to a goal-based criterion, denoted $g(c)$, is defined as follows. If c is already a goal, then $g(c) = c$. Otherwise:

- For every $x \in Dom(X)$, define a goal c_x on variable $X = x$ with $\top \trianglerighteq_{c_x} \bot$.
- Define a lexicographic criterion $g(c) = \langle C_{g(c)}, \trianglerighteq_{g(c)} \rangle$ such that $C_{g(c)} = \{c_x \mid x \in Dom(x)\}$ and $c_x \trianglerighteq_{g(c)} c_{x'}$ iff $x \trianglerighteq_c x'$.

Example 4.13. To illustrate, Figure 4.8 displays the translation of the criterion tree in Figure 4.7. The simple criteria C and D have been replaced by their translations $g(C)$ and $g(D)$. These lexicographic criteria have a subgoal for every value of C resp. D. The priority between these goals corresponds to the value preferences of the original simple criteria.

Theorem 4.4. Let $c = \langle X, \trianglerighteq \rangle$ be a simple criterion such that $Dom(X_c)$ is finite. The goal-based translation $g(c)$ of c as defined in Definition 4.16 is equivalent to c.

Proof. We distinguish five possible cases and show that in every case, c's preference between α and β is the same as $g(c)$'s preference between α and β.

1. If $\alpha_X = \beta_X$ then (a) $\alpha \approx_c \beta$ and (b) $\alpha \approx_{g(c)} \beta$.
2. If $\alpha_X \doteq_c \beta_X$ but $\alpha_X \neq \beta_X$ then (a) $\alpha \approx_c \beta$ and (b) $\alpha \approx_{g(c)} \beta$.
3. If $\alpha_X \gtrdot_c \beta_X$ then (a) $\alpha \succ_c \beta$ and (b) $\alpha \succ_{g(c)} \beta$.
4. If $\beta_X \gtrdot_c \alpha_X$ then (a) $\beta \succ_c \alpha$ and (b) $\beta \succ_{g(c)} \alpha$.
5. If $\alpha_X \not\gtrdot_c \beta_X$ and $\beta_X \not\gtrdot_c \alpha_X$ then (a) $\alpha \not\succeq_c \beta$ and $\beta \not\succeq_c \alpha$ and (b) $\alpha \not\succeq_{g(c)} \beta$ and $\beta \not\succeq_{g(c)} \alpha$.

1-5(a). This follows directly from the definition of simple criteria. **1(b).** If $\alpha_X = \beta_X$ then $\forall x \in Dom(X) : \alpha_{X=x} = \beta_{X=x}$, so also $\forall x \in Dom(X) : \alpha \approx_{c_x} \beta$. Hence, by the definition of a lexicographic criterion: $\alpha \approx_{g(c)} \beta$. **2-5(b).** If $\alpha_X \neq \beta_X$ then $\forall x \in Dom(X) \backslash \{\alpha_X, \beta_X\} : \alpha_{X=x} = \beta_{X=x}$ and $\alpha \approx_{g(c)} \beta$. Since a subcriterion s of a compound criterion such that $\alpha \approx_s \beta$ does not influence that compound criterion's preference between α and β, the only criteria that can influence $g(c)$'s preference between α and β are c_{α_X} and c_{β_X}. Since $\alpha \succ_{c_{\alpha_X}} \beta$ and $\beta \succ_{c_{\beta_X}} \alpha$, preference between α and β by $g(c)$ is determined by the priority between c_{α_X} and c_{β_X}. **2(b).** If $\alpha_X \geq_c \beta_X$ then $c_{\alpha_X} \triangleq_{g(c)} c_{\beta_X}$, so they are together in a cardinality criterion and we have $\alpha \approx_{g(c)} \beta$. **3(b).** If $\alpha_X \gtrdot \beta_X$ then $c_{\alpha_X} \triangleright_{g(c)} c_{\beta_X}$ so by the definition of a lexicographic criterion $\alpha \succ_{g(c)} \beta$. **4(b).** Analogous to 3(b). **5(b).** If $\alpha_X \not\gtrdot_c \beta_X$ and $\beta_X \not\gtrdot_c \alpha_X$ then $c_{\alpha_X} \ntrianglerighteq_{g(c)} c_{\beta_X}$ and $c_{\beta_X} \ntrianglerighteq_{g(c)} c_{\alpha_X}$ and $c_{\alpha_X} \not\triangleq_{g(c)} c_{\beta_X}$, so by the definition of a lexicographic criterion $\alpha \not\succeq_{g(c)} \beta$ and $\beta \not\succeq_{g(c)} \alpha$. □

By replacing every simple criterion c in a criterion tree with its goal-based translation $g(c)$, an equivalent goal-based criterion tree is obtained.

Definition 4.17. (Relative succinctness) c' is *at least as succinct* as c iff there exists a polynomial function p such that $size(c') \leq p(size(c))$. (Adapted from [41].)

Theorem 4.5. Let $c = \langle X_c, \geq_c \rangle$ be a simple criterion such that $Dom(X_c)$ is finite. The translation $g(c)$ of c as defined in Definition 4.16 is just as succinct as c.

Proof. The goal-based translation just replaces variable values with goals, and the preference relation between them with an identical priority relation between goals, so the translation is linear. □

The above two theorems are very important as they show that goals are very expressive as a way to represent qualitative preferences, and moreover, that this representation is just as succinct as a representation based on multi-valued criteria.

4.4.3 Satisfaction level goals

We have shown that simple criteria based on multi-valued variables can be translated to equivalent goal-based criteria. Note that this is not the same as restricting the domain of all variables to be Boolean, as is done in some other approaches (e.g. [33]). While goals are Boolean, the underlying structure of the outcomes remains the same. One advantage of this is that goals can be chosen independently from the factual representation of outcomes.

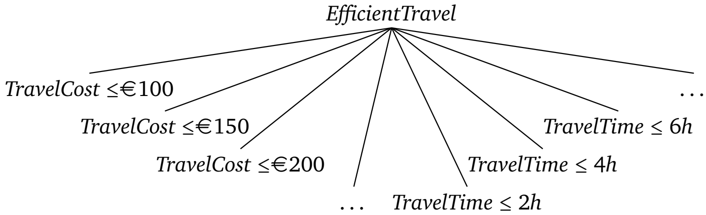


Figure 4.9: Representing preferences with satisfaction level goals

When we translate a multi-valued simple criterion into a goal-based lexicographic criterion, every subcriterion is a goal of the type $X = v$ where X is the original simple criterion's variable and v is one of its possible values. Preference between the possible values of X is translated to priority between goals. However, if the preference between possible values follows a natural ordering on the values, such as $\leq$ or $\geq$ on natural numbers, there is another intuitive possibility. Instead of specifying goals with equations, we can also use other comparisons such as inequalities.

Consider for example a variable X with three levels of satisfaction: $Dom(X) = \{1, 2, 3\}$ and $3 > 2 > 1$. Now instead of goals $X = 3$, $X = 2$ and $X = 1$, we specify goals $X \geq 3$, $X \geq 2$ and $X \geq 1$. We call such goals *satisfaction level goals*. These goals can be combined in a compound criterion. If just the goals relating to a single variable are combined, it does not matter what kind of compound criterion (lexicographic or cardinality) or what priority ordering is chosen; the resulting preference relation will be the same. This is due to the special relation between satisfaction level goals relating to the same variable: when one of them prefers α over β all others either also prefer α over β or are indifferent between them, but none can prefer β over α. Of course, when satisfaction level goals relating to different variables are combined, the choice of compound criterion and priority does make a difference.

Satisfaction level goals provide a straightforward way to specify goals on continous domains by discretizing them. Moreover, they provide the means to compensate between different variables in a qualitative way. It may be worth noting that if all combined variables share the same discrete domain and a satisfaction level goal is specified for every value, this approach coincides with taking the sum of all variables.

Example 4.14. Efficient travel is a tradeoff between the travel time and the cost. Both of these variables are measured on a (nearly) continous scale of hours and euros respectively. These domains can be discretized by defining satisfaction level goals on them. By varying the distance between satisfaction levels, the compensation between time and cost can be fine-tuned. An example is shown in Figure 4.9.

4.5 Updates in a QPS

In this section we show that goal-based criterion trees also have some added value compared to trees with multi-valued criteria. We introduce updates on a criterion tree as changes in the value preference of simple criteria or in the priority of lexicographic criteria. The number of possible updates of this kind depends on the structure of the tree. In general, the flatter a criterion tree, the more updates are possible. It is possible to make criterion tree structures flatter, i.e. to reduce the depth of the tree, by removing intermediate lexicographic criteria. The advantage of goal-based criterion trees is that they can be flattened to a greater extent than their equivalent non-goal-based counterparts. We first formalize the concept of flattening a criterion tree. Then we define what we mean by basic updates in a criterion tree and show the advantages of flat goal-based QPSs compared to other flat QPSs.

4.5.1 Flattening

Simple criteria are terminal nodes (leaves) and cannot be flattened further. We assume that cardinality criteria are always goal-based, which means they have only goals as subcriteria and cannot be flattened either. Lexicographic criteria can have three kinds of subcriteria: simple, cardinality and lexicographic. They can be flattened by replacing each lexicographic subcriterion by that criterion's subcriteria and adapting the priority accordingly, as defined in the following definition.

Definition 4.18. (Removing a lexicographic subcriterion) Let $c = \langle C_c, \rhd_c \rangle$ be a lexicographic criterion and $d = \langle C_d, \rhd_d \rangle \in C_c$ a lexicographic criterion that is a subcriterion of c. We now define a lexicographic criterion $f(c,d) = \langle C_{f(c,d)}, \rhd_{f(c,d)} \rangle$ that is equivalent to c but does not have d as a subcriterion. To this end, we define $C_{f(c,d)} = C_c \backslash \{d\} \cup C_d$ and $\forall i,j \in C_{f(c,d)} : i \rhd_{f(c,d)} j$ iff

- $i,j \in C_c$ and $i \rhd_c j$, or
- $i,j \in C_d$ and $i \rhd_d j$, or
- $i \in C_c$, $j \in C_d$ and $i \rhd_c d$, or
- $i \in C_d$, $j \in C_c$ and $d \rhd_c j$.

Theorem 4.6. $f(c,d)$ is equivalent to c, i.e. $\alpha \succeq_c \beta$ iff $\alpha \succeq_{f(c,d)} \beta$.

Proof. $\Rightarrow$. Suppose $\alpha \succeq_c \beta$. Then $\forall s \in C_c(\alpha \succeq_s \beta \vee \exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \rhd_c s))$. We need to show that also $\forall s \in C_{f(c,d)}(\alpha \succeq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha \succ_{s'} \beta \wedge s' \rhd_{f(c,d)} s))$. We do this by showing that $\alpha \succeq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha \succ_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ holds for every possible origin of $s \in C_{f(c,d)}$. We have $\forall s \in C_{f(c,d)}$, either $s \in C_c \backslash \{d\}$ or $s \in C_d$.

- If $s \in C_c \backslash \{d\}$, we know that $\alpha \succeq_s \beta \vee \exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \rhd_c s)$.
    - If $\alpha \succeq_s \beta$, trivially also $\alpha \succeq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha \succ_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ and we are done.
    - If $\exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \rhd_c s)$, then either $s' \in C_c \backslash \{d\}$ or $s' = d$.
        - If $s' \in C_c \backslash \{d\}$, then $s' \in C_{f(c,d)}$ and $s' \rhd_{f(c,d)} s$, so also $\alpha \succeq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha \succ_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ and we are done.

- If $s' = d$, then (since $\alpha >_{s'} \beta$) $\exists i \in C_{s'}$ (and hence $\in C_{f(c,d)}$): $\alpha >_i \beta$. Since $s' \rhd_c s$, we have $i \rhd_{f(c,d)} s$ and so also $\alpha \geq_s \beta \vee \exists i \in C_{f(c,d)}(\alpha >_i \beta \wedge i \rhd_{f(c,d)} s)$ and we are done.

- Now consider the case that $s \in C_d$. Since $d \in C_c$, we know that either $\alpha \geq_d \beta$ or $\exists s' \in C_c(\alpha >_{s'} \beta \wedge s' \rhd_c d)$.
  - If $\alpha \geq_d \beta$, we know $\alpha \geq_s \beta \vee \exists s' \in C_d(\alpha >_{s'} \beta \wedge s' \rhd_d s)$ and hence $\alpha \geq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha >_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ and we are done.
  - If $\exists s' \in C_c(\alpha >_{s'} \beta \wedge s' \rhd_c d)$ then $\exists s' \in C_{f(c,d)}(\alpha >_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ so trivially also $\alpha \geq_s \beta \vee \exists s' \in C_{f(c,d)}(\alpha >_{s'} \beta \wedge s' \rhd_{f(c,d)} s)$ and we are done.

$\Leftarrow$. Suppose $\alpha \ngeq_c \beta$. Then $\exists s \in C_c(\alpha \ngeq_s \beta \wedge \forall s' \in C_c(s' \rhd_c s \to \alpha \not>_{s'} \beta))$. We need to show that also $\exists t \in C_{f(c,d)}(\alpha \ngeq_t \beta \wedge \forall t' \in C_{f(c,d)}(t' \rhd_{f(c,d)} t \to \alpha \not>_{t'} \beta))$. Either $s \neq d$ or $s = d$.

- If $s \neq d$, then $s \in C_{f(c,d)}$ and we know that $\alpha \ngeq_s \beta$ and $\forall s' \in C_{f(c,d)} \backslash C_d(s' \rhd_{f(c,d)} s \to \alpha \not>_{s'} \beta)$.
  - If $d \not\rhd_c s$, then $\forall s' \in C_{c*}(s' \rhd_{f(c,d)} s \to s' \in C_{f(c,d)} \backslash C_d)$. So we have $\exists s \in C_{f(c,d)}(\alpha \ngeq_s \beta \wedge \forall s' \in C_{f(c,d)}(s' \rhd_{f(c,d)} s \to \alpha \not>_{s'} \beta))$. Take $t = s$ and we are done.
  - If $d \rhd_c s$, then $\alpha \not>_d \beta$, i.e. $\alpha \ngeq_d \beta$ or $\beta \geq_d \alpha$.
    - If $\alpha \ngeq_d \beta$, then $\exists u \in C_d(\alpha \ngeq_u \beta \wedge \forall u' \in C_d(u' \rhd_d u \to \alpha \ngeq_{u'} \beta))$. Since $\forall s' \in C_c(s' \rhd_c s \to \alpha \not>_{s'} \beta)$ and $d \rhd_c s$, we also have $\exists u \in C_{f(c,d)}(\alpha \ngeq_u \beta \wedge \forall u' \in C_{f(c,d)}(u' \rhd_{f(c,d)} u \to \alpha \not>_{u'} \beta))$. Take $t = u$ and we are done.
    - If $\beta \geq_d \alpha$, then $\forall v \in C_d(\beta \geq_v \alpha \vee \exists v' \in C_d(\beta >_{v'} \alpha \wedge v' \rhd_d v))$. This means that either $\forall u \in C_d(\beta \geq_u \alpha)$ or $\exists u \in C_d(\beta >_u \alpha \wedge \neg\exists u' \in C_d(u' \rhd_d u))$.
      - If $\forall u \in C_d(\beta \geq_u \alpha)$, then $\forall u \in C_d(\alpha \not>_u \beta)$. Take $t = s$ and we are done.
      - If $\exists u \in C_d(\beta >_u \alpha \wedge \neg\exists u' \in C_d(u' \rhd_d u))$, then $\exists u \in C_d(\alpha \ngeq_u \beta \wedge \forall u' \in C_d(u' \rhd_d u \to \alpha \not>_{u'} \beta))$. Take $t = u$ and we are done.

- If $s = d$, then $\alpha \ngeq_d \beta$, so $\exists u \in C_d(\alpha \ngeq_u \beta \wedge \forall u' \in C_d(u' \rhd_d u \to \alpha \not>_{u'} \beta))$. Since $\forall s' \in C_c(s' \rhd_c d \to \alpha >_{s'} \beta)$, we have $\forall s' \in C_c(s' \rhd_c u \to \alpha >_{s'} \beta)$. Take $t = u$ and we are done. □

Theorem 4.7. $f(c,d)$ is just as succinct as c.

Proof. When a lexicographic subcriterion is removed according to Definition 4.18, the total number of criteria decreases by 1: the subcriteria of d become direct subcriteria of c and d itself is removed. The priority between the original subcriteria of c (i.e. $C_c \backslash \{d\}$) and the priority between the original subcriteria of d (i.e. C_d) remains unaltered. Just the priority between the subcriteria in $C_c \backslash \{d\}$ and d is replaced by priority between the subcriteria in $C_c \backslash \{d\}$ and the subcriteria in C_d. Since $|C_d|$ is finite, the increase in size is linear. □

Definition 4.19. (Flat criterion) A criterion is *flat* iff it is (i) a simple criterion, (ii) a goal-based cardinality criterion, or (iii) a lexicographic criterion such that all its subcriteria are either simple criteria or goal-based cardinality criteria.

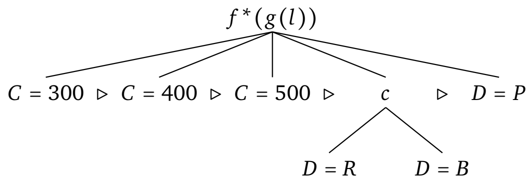


Figure 4.10: The result of flattening the criterion in Figure 4.8

Definition 4.20. (Flattening) The flat version of a non-flat lexicographic criterion c, denoted $f^*(c)$, is obtained as follows. For an arbitrary lexicographic subcriterion $d \in C_c$, get $f(c,d)$. If $f(c,d)$ is flat, $f^*(c) = f(c,d)$. Otherwise, $f^*(c) = f^*(f(c,d))$.

Example 4.15. The original criterion tree in Figure 4.7 is already flat. Its goal-based translation in Figure 4.8 can be flattened further, as shown in Figure 4.10. Here the lexicographic subcriteria $g(C)$ and $g(D)$ have been removed.

4.5.2 Updates

Criterion trees can be updated by leaving the basic structure of the tree intact but changing the priority ($\trianglerighteq$) between (in)direct subcriteria of a lexicographic criterion (in the alternative specification) or the value preferences of a multi-valued simple criterion ($\geq$). By performing these basic operations, the induced preference relation also changes. Therefore, such updates can be used to 'fine-tune' a person's preference representation.

Definition 4.21. (Update) An *update* of a criterion tree is a change in (i) the preference between values ($\geq$) of a multi-valued simple criterion; and/or (ii) the priority ($\trianglerighteq$) between (in)direct subcriteria of a lexicographic criterion (in the alternative specification). The changed relations still have to be preorders.

Translating a criterion to a goal-based criterion does not change the updates that can be performed.

Theorem 4.8. For every update on a criterion tree c, there exists an equivalent update on the goal-based translation $g(c)$ and vice versa.

Proof. Every change in a value preference $\geq$ between two values x and y corresponds one-to-one to a change in priority between c_x and c_y. Every change in priority between two subcriteria s and s' corresponds one-to-one to a change in priority between $g(s)$ and $g(s')$. □

Example 4.16. Consider for example the criterion tree in Figure 4.7. On the highest level, there are three possibilities for the priority: $C \rhd D$, $D \rhd C$ or incomparable priority. On the next level, each simple criterion has preferences over three possible

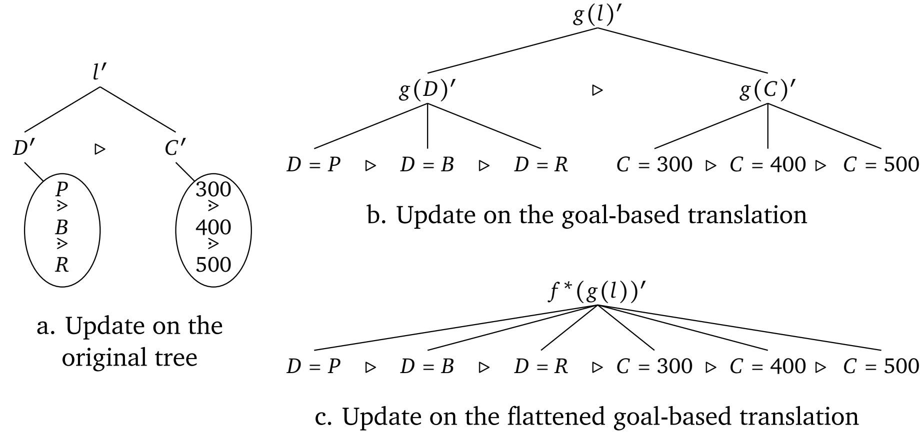


Figure 4.11: Updates on criterion trees

values, which can be ordered in 29 different ways (this is the number of different preorders with three elements, oeis.org/A000798). So in total there are $3 \times 29 \times 29 = 2523$ possible updates of this tree. For the goal-based translation of this tree (in Figure 4.8) this number is the same. Figure 4.11a shows one alternative update of the original criterion tree in Figure 4.7; Figure 4.11b shows the same update on its goal-based translation in Figure 4.8.

Flattening a criterion tree influences the updates that can be performed; all updates that are possible on the non-flat tree can also be performed on the flattened version, but not vice versa. That is, flattening a criterion tree introduces more possible updates.

Theorem 4.9. For every update on a criterion tree c, there exists an equivalent (set of) update(s) on the flattened criterion tree $f^*(c)$.

Proof. Since simple criteria are not altered in the flattening process, every change in a value preference $\geq$ between two values x and y can also be applied in the flattened version. Every change in priority between two subcriteria s and s' corresponds to a change in priority between all of the (in)direct subcriteria of s that are flat and all of the (in)direct subcriteria of s' that are flat. □

Example 4.17. Figure 4.11c shows an update on the flat goal-based criterion tree in Figure 4.10 that is equivalent to the updates in Figure 4.11a and 4.11b.

Theorem 4.10. If a criterion tree c is not flat, there exist updates on $f^*(c)$ that do not have equivalent updates on c.

We show this by means of an example.

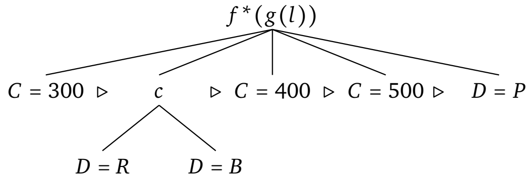


Figure 4.12: Alternative flat goal-based tree obtained by updating the tree in Figure 4.10

Example 4.18. The goal-based tree in Figure 4.8 can be flattened to the equivalent flat tree in Figure 4.10. This flattened tree can be updated in 209527 different ways (the number of different preorders with 6 elements, oeis.org/A000798), thereby allowing more preference relations to be represented by the same tree structure. Figure 4.12 shows an alternative flat goal-based tree that can be obtained from the previous one by updating it. It is not possible to obtain an equivalent criterion tree by fine-tuning the original criterion tree or its goal-based translation. This is because goals relating to different variables are 'mixed': the most important goal is that the cost is 300, the next most important goal is that the destination is Rome or Barcelona, and only after that is the cost considered again. This is not possible in a criterion tree that is based on simple criteria that are defined directly on the variables C and D.

We have seen that the same updates are possible on a multi-valued criterion tree and its goal-based translation. If, however, both trees are flattened, more updates are possible on the flattened goal-based tree.

Theorem 4.11. Let c be a non-goal-based criterion. Then there exist updates on $f^*(g(c))$ that do not have equivalent updates on $f^*(c)$.

In general, the flatter a criterion tree, the more different updates are possible (Theorem 4.10). Since a goal-based tree can be made flatter than an equivalent criterion tree that is based on multi-valued simple criteria, the goal-based case allows more updates. This is visualized in Figure 4.13.

4.5.3 Fine-tuning

The results above show that every update that can be applied on a criterion tree can also be applied on its flattened goal-based translation, and that this last criterion tree even allows more updates. However, if we look at the size of the updates, we can see that for equivalent updates, more value preference or priority relations have to be changed when the structure is flatter. For example, a simple inversion of the priority between $g(C)$ and $g(D)$ in Figure 4.8 corresponds to the inversion of priority between all of $C = 300$, $C = 400$ and $C = 500$ and all of $D = R$, $D = B$ and $D = P$ in Figure 4.10. This suggests the following approach to fine-tuning a given preference

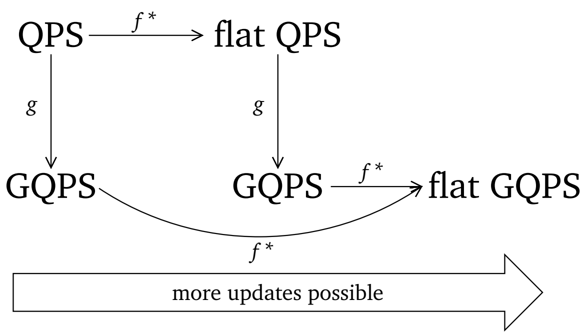


Figure 4.13: Effects of goal-based translation and flattening on possible updates

representation during the preference elicitation process. First, one can fine-tune the current criterion tree as well as possible using (coarse) updates. If the result does not match the intended preferences well enough, one can start flattening, which will create more, fine-grained possibilities to update the tree. If this still does not allow to express the correct preferences, one can make a goal-based translation and flatten it. This allows for even more possible updates on an even lower level.

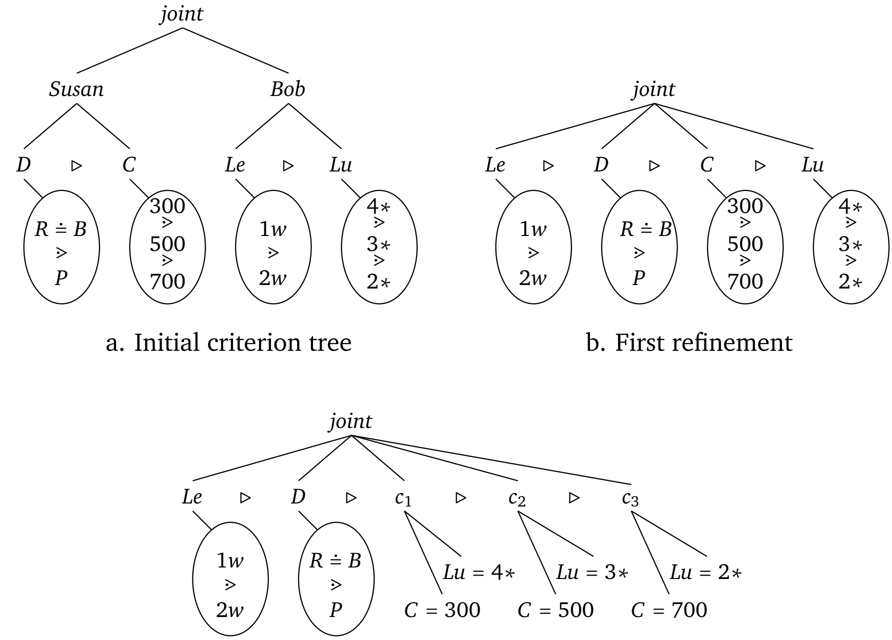
Example 4.19. Susan and Bob are planning a city trip together. Susan would like to go to a city that she has not been to before, and hence prefers Rome or Barcelona to Paris as destination (D). She also does not want to spend too much money, so she prefers a low cost (C). Bob is a busy businessman who only has a single week of holiday, so he prefers a shorter length (Le), and would like some luxury (Lu), expressed in the number of stars of the hotel. There is no priority between Susan's and Bob's preferences. The initial criterion tree for Susan and Bob's joint preferences is displayed in Figure 4.14a.

Susan and Bob decide that Bob's criterion on the length of the trip should be the most important, because he really does not have time to go for two weeks. They also decide that luxury is less important than the other criteria. In order to update the tree, it is first flattened by removing the subcriteria of Susan and Bob. The new tree, after flattening and updating, is shown in Figure 4.14b.

However, Bob feels that luxury can compensate for cost. To represent this, the criteria for cost and number of stars are translated to goals and combined into three cardinality criteria, as shown in Figure 4.14c. At this point, the travel agent's website is able to make a good selection of offers to show and recommend to Susan and Bob.

4.6 Related work

Gérard *et al.* [56] propose a qualitative approach for ordering alternatives, which are represented as a vector of qualitative criteria evaluations (satisfaction levels). In



a. Initial criterion tree b. First refinement

c. Second refinement

Figure 4.14: Successive criterion trees for Susan and Bob

contrast to our approach, they do not determine preferences by aggregating the preference orderings given by each criterion, but define preference between alternatives as a total preorder that satisfies all given constraints. A constraint can be a specific preference statement ('outcome α is preferred to outcome β'), or a generic preference such as Pareto ordering. Another difference is that they use the same linearly ordered scale of satisfaction levels for every criterion, whereas we allow arbitrary domains for variables and arbitrary preorders for preference among values from those domains.

Qualitative choice logic [34] adds a new operator called *ordered disjunction* to propositional logic. Benferhat and Sedki [20] extend QCL in order to handle negated and conditional preferences. The semantics of QCL and the extensions is based on the degree of satisfaction of 'choice formulas'; the more formulas are satisfied to a high degree, the better the alternative. In contrast, QPS uses explicit criteria which do not use the same common satisfaction levels, but can represent any preorder between alternatives.

Wilson [136] introduces an extension of CP-nets based on preference statements of the form 'low price is preferred over high price *irrespective of the values of other variables*'. It is possible to represent CP-nets in this logic but the logic goes beyond

CP-nets and is also able to represent lexicographic orderings. One of the main differences with our framework is that a QPS in addition supports abstraction (conceptual knowledge). Abstraction allows us to represent certain orderings that cannot be represented in Wilson's logic. An example is the 'battle of the sexes' that was discussed above. The ordering $AB > \neg A\neg B > A\neg B > \neg AB$ cannot be represented by Wilson's logic, as there is no worsening swap between (A, B) and $(\neg A, \neg B)$. By introducing a new variable T which is defined by $T = A \leftrightarrow B$ this order can be represented in a QPS, as was shown above.

Van der Weide [133] proposes a conceptual framework and argumentation system for reasoning about the values of a decision maker: the Perspective-based Value Model (PVM) and Argumentation System for Perspective-based Value (ASPV). In this framework, the preferences of a decision maker are influenced by his values, which are conceptually similar to our notion of underlying interests. Every value provides a perspective on the goodness of options, and perspectives may positively or negatively influence other perspectives. For example, if an agent A has a value V and one outcome α is better than another outcome β from the perspective of V, then this is a (defeasible) reason for a preference of α over β by agent A. These influences are modelled in a so-called Perspective-based Value Comparison Structure (PVCS). Influences can be positive ($p \uparrow q$: p positively influences q) or negative ($p \downarrow q$: p negatively influences q), and are transitive. However, it is not specified how influences interact when they contradict each other. For example, suppose that α is better than β from perspective p, β is better than α from perspective q, and both p and q positively influence perspective r. Then we have a reason (argument) to prefer α to β from perspective r, but also a reason (argument) to prefer β to α from perspective r. The two arguments attack each other. Van der Weide proposes that this conflict should be resolved by meta-argumentation, accruing arguments with the same conclusion and taking the relative strengths of arguments into account, but does not discuss the exact procedure. This approach is inherently different from the approach taken by Qualitative Preference Systems. In a QPS, all preferences are ultimately based on preferences specified between the possible values of variables (simple criteria). In contrast, the preferences of the perspectives that are lowest in a PVCS are assumed to be given. Also, when a QPS compound criterion has multiple subcriteria, its resulting preference is unequivocally defined even when the subcriteria contradict each other. In [128], an argumentation framework was presented for reasoning about a QPS and the properties of outcomes. Unfortunately, since the meta-argumentation framework of Van der Weide is not completely specified, we cannot give a detailed comparison between the two approaches.

Amgoud and Prade [7] propose a bipolar approach where a distinction is made between goals and rejections. In the multi-criteria decision making setting, criteria are mapped to a bipolar scale $T = \{-k, \ldots, -1, 0, +1, \ldots, +k\}$ (which can be seen as a generalisation of the three levels of value used by [133]). For every criterion c_i and every level j on the positive part of T, a goal $g_i^j : c_i \geq +j$ is specified. Similarly, for every criterion c_i and every level j on the negative part of T, a rejection $r_i^j : c_i \leq -j$ is specified. The importance of goals is defined as $k-j+1$, for rejections the importance is j. If criteria have different level of importance, the importance of a goal (resp. rejection) is the minimum of $k-j+1$ (resp. j) and the importance of the associated criterion. Decisions are compared with decision principles, which were discussed in Section 4.3.3. In [7], it is assumed that all criteria use the same totally ordered bipolar scale for the degree of satisfaction (criteria are *commensurate*). Importance of arguments is also measured on the same scale. In contrast, in a QPS the preference relations of different criteria are independent (no commensurability) and can be any (possibly partial) preorder.

4.7 Conclusion

We introduced Qualitative Preference Systems, a new framework for representing qualitative multi-criteria preferences. QPSs combine different features for compactly expressing preferences. These features include the well-known lexicographic rule which combines basic preferences over variables, and a cardinality-based rule which counts criteria that are satisfied. In addition, QPSs enable the representation of knowledge, which allows for expressing feasibility constraints as well as abstractions (concept definitions). Finally, such systems support a layered structure for representing preference orderings. This combination of features provides an expressive preference representation framework which at the same time allows for a compact representation of preference orderings.

We have discussed the issue of preferential dependence between attributes. We illustrated how conditional preferences can be modelled in a QPS, but argued that modelling underlying interests instead would be a more natural solution. We illustrated several ways to model interests in a QPS and showed how background knowledge, which can be used to express constraints and define abstract concepts, is essential in such situations as it can be used to specify criteria on a more fundamental level. We have shown that the QPS framework can be used to model preferences between outcomes based on goals. It has several advantages over other approaches. First, the QPS framework is general and flexible and can model several interpretations of using goals to derive preferences between outcomes. This is done by simply adapting the structure of the criterion tree. It is possible to specify an incomplete preference relation such as the ceteris paribus relation by using an incomplete priority ordering. But if a complete preference relation is needed, it is also easy to obtain one by completing the priority relation between subcriteria of a lexicographic criterion, or using cardinality criteria. Second, goals do not have to be independent. Multiple goals can be specified using the same variable. For example, there is no problem in specifying both p and $p \wedge q$ as a goal. Third, goals do not have to be consistent. It is not contradictory to have both p preferred to $\neg p$ (from one perspective) and $\neg p$ preferred to p (from another). This possibility is also convenient when combining preferences of multiple agents, who may have different preferences. Preferences of multiple agents can be combined by just collecting them as subcriteria of a new lexicographic criterion. The straightforward representation of goals in the QPS framework also applies to bipolar goals. Both achievement goals (whose satisfaction is an argument pro) and avoidance goals (whose non-satisfaction is an argument con) can be modelled as QPS goals, and combined together using compound criteria.

We have shown that the Logical Preference Descriptions introduced in [33] can be embedded in the QPS framework, with the exception of the disjunction operator which is not natural as it does not satisfy independence of irrelevant alternatives and unanimity with abstentions [8]. The 'logical' operators of [33] translate to structural features of QPSs. We have also shown that QPSs are able to express conditional preferences by providing an order preserving embedding of acyclic CP-nets into QPSs which satisfies the ceteris paribus condition. Last but not least, these embeddings are size preserving, i.e. the resulting QPSs provide a representation that is as succinct as the LPD or CP-net representation. This fact indicates that various problems such as dominance testing for QPSs have an associated computational complexity that is at most as difficult as these alternative frameworks for preference representation.

Preferences are usually based on orderings of the possible values of each variable, which can be Boolean, numeric, or nominal. We have shown that multi-valued criteria can be translated to equivalent goal-based criteria. Such a translation requires at most polynomially more space, and hence is just as succinct as the original QPS. This result shows that goals are very expressive as a representation of qualitative preferences among outcomes.

Goal-based criterion trees also have some added value compared to trees with multi-valued criteria. We introduced basic updates on a QPS and showed that goal-based QPSs allow for more fine-grained updates than their multi-valued counterparts. This is due to the different structure of goal-based criteria. In general, the flatter a criterion tree, the more updates are possible. It is possible to make criterion tree structures flatter, i.e. to reduce the depth of the tree, by removing intermediate lexicographic criteria. The advantage of goal-based criterion trees is that they can be flattened to a greater extent than their equivalent non-goal-based counterparts, and hence provide more possible updates. We proposed a procedure to fine-tune a criterion tree during the preference elicitation process. Essentially, this is a top-down approach where a criterion tree is first updated as well as possible in its current state, and is only flattened and/or translated to a goal-based tree if more updates are necessary. This procedure gives rise to a more fundamental question. If it is really necessary to take all these steps, then maybe the original criteria were not chosen well in the first place. It may have been better to choose more fundamental interests as criteria.

Chapter 5

An argumentation framework for qualitative multi-criteria preferences

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.

5.1 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 common-sense 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) in the sense of [2]. In a PAF, preferences 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 [72]. 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 5.2, we briefly recall Qualitative Preference systems. Section 5.3 presents the argumentation framework that provides the means to reason about a QPS. In Section 5.4 we extend the argumentation framework with background knowledge and the means to derive information by default. Finally, Section 5.5 concludes the paper.

5.2 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 5.1. (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 5.2. (Simple criterion) A *simple criterion* c is a tuple $\langle X_c, \trianglerighteq_c \rangle$, where $X_c \in Var$ is a variable, and $\trianglerighteq_c$, a preference relation on the possible values of X_c, is a preorder on $Dom(X_c)$. A simple criterion $c = \langle X_c, \trianglerighteq_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $\alpha_{X_c} \trianglerighteq_c \beta_{X_c}$.

Definition 5.3. (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 $\alpha \succeq_c \beta$, iff $|\{s \in C_c \mid \alpha \succ_s \beta\}| \geq |\{s \in C_c \mid \alpha \not\succeq_s \beta\}|$.

Definition 5.4. (Lexicographic criterion) A *lexicographic criterion* c is a tuple $\langle C_c, \triangleright_c \rangle$, where C_c is a nonempty set of criteria (the *subcriteria* of c) and $\triangleright_c$, 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 $\alpha \succeq_c \beta$, iff $\forall s \in C_c(\alpha \succeq_s \beta \vee \exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \triangleright_c s))$.

5.3 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.

5.3.1 Abstract argumentation framework

Our argumentation framework is a concrete instantiation of an abstract argumentation framework as defined by Dung [52]. To define which arguments are justified, we use Dung's preferred semantics.

Definition 5.5. (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 5.6. (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 [104] 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.

5.3.2 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. [131, 104]).

Definition 5.7. (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.

5.3.3 Defeat

We define two different kinds of defeat: rebuttal and undercut (note that, unlike e.g. [104], 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. [104]. 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 5.8. (Defeat) An argument A *defeats* an argument B ($A \to B$) 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.

5.3.4 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 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 5.1 and 5.2.

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 5.4 in order to reason with background knowledge.

Definition 5.9. (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 $\in \mathcal{P}_{out}$
$literal$	::=	$literal_{in} \mid atom_{out}$
$formula$	::=	$literal \mid \sim literal$

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

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

Table 5.1: The predicates in $\mathcal{P}_{in}$ and their interpretation

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

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.

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 5.2: The predicates in $\mathcal{P}_{out}$ and their interpretation

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 5.1. 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)
```

5.3.5 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.

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

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.

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

Example 5.2. The following argument infers that simple criterion `c1` prefers `o1` over `o2`. Similar arguments can be constructed for `c2` and `c3`.

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

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.

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

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.

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

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 5.3. The following argument concludes that there is one subcriterion of c4 that strictly prefers o1 over o2.

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

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

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

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

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.

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

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

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. $\texttt{pref}(c, o_1, o_2)$ 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 $\texttt{pref}(c, o_1, o_2)$, 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 5.4. The following argument concludes that `c5` prefers `o1` to `o2` based on its subcriterion `c4`.

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

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`.

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

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

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

5.3.6 Correspondence between QPS and AF

Theorem 5.1. 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.

5.4 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 corresponds 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.

5.4.1 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 is used by Prakken and Sartor [105]).

Definition 5.10. (Language) The *input language* is defined as follows.

$atom_{in}$	::=	$p(t_1,\ldots,t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{in}$
$atom_K$	::=	$p(t_1,\ldots,t_n)$ where p is an n-ary predicate $\in \mathcal{P}_K$
$literal_{in}$	::=	$atom_{in} \mid atom_K \mid \neg atom_K$
$rule$	::=	$literal_{in}$, ..., $literal_{in}$, ~ $literal_{in}$, ..., ~ $literal_{in}$ => $literal_{in}$
$formula_{in}$	::=	$literal_{in} \mid rule$

The *full language* is defined as follows.

$atom_{out}$	::=	$p(t_1,\ldots,t_n)$ where p is an n-ary predicate $\in \mathcal{P}_{out}$
$literal$	::=	$literal_{in} \mid atom_{out}$
$formula$	::=	$literal$ | ~ $literal$ | $rule$

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 5.5. 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)
```

5.4.2 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 5.6. Below are some of the arguments that can be built with the knowledge base from Example 5.5. 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).`

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

The argument deriving a preference for `o1` over `o2` by criterion `c5` is the same as in Example 5.4, 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).

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

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

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

5.5 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* [94, 7]. 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, Amgoud and Prade [7] 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 [26, 50], 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* [17, 16]. Values are used in the sense of 'fundamental social or personal goods that are desirable in themselves' [16], 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. Kaci and Van der Torre [71] 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. Van der Weide *et al.* [134] 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.

Chapter 6

Explaining qualitative preference models

Abstract We propose an explanation facility for a qualitative preference representation framework. We show how an explanation can be provided for qualitative, multi-criteria preferences based on the criteria that are used to decide preferences between outcomes. Such a facility provides an important tool for a user to understand how preferences are determined. We show that this facility can also be used by a user to inform the system about its preferences. Such a user-provided explanation can be used for updating and improving a preference model maintained by the system.

6.1 Introduction

A preference representation framework provides a tool for determining preferences between outcomes. That is, for any two outcomes it can determine whether one is strictly preferred to the other, both are equally preferred, or they are incomparable. In this paper, we discuss an additional facility, namely the *explanation* of preferences maintained by such a system. Explanation of preferences is useful and important in many cases, such as situations where a decision maker has to explain his decision to other actors; where a decision support system that is elicited from an expert has to explain its list of recommended options to a non-expert user; or where agents may give each other feedback on offers in negotiation, without revealing all their preferences [82]. Another reason to use explanation is to improve users' confidence in a system, since lack of confidence is an obstacle to acceptance and practical use of the system [92]. In these cases, it is not satisfactory to just present the preference model. Although this model does contain all information on which the preference is based, the format is not suitable for presentation to a user. First, the model is too technical for the average human user to interpret. Second, even experts may have trouble interpreting the model since it may be quite large, and hence it would be hard to quickly find the reason behind the preference.

Besides explaining someone's preferences to another party, explanation may also be used 'in reverse' during preference elicitation and updating. Here the idea is as follows. The user is not only asked to state his preference between two given outcomes, but also to explain this preference. This explanation can then be used to update the preference model in such a way that the explanation for the user's preference that would be generated by the updated model coincides with the explanation given by the user.

In this paper we propose an approach to generate explanations from a preference model and to use explanations to update a preference model. The preference models we consider are expressed in a particular preference representation framework called Qualitative Preference Systems (QPS) [121, 130]. QPS is a general framework for the representation of qualitative, multi-criteria preferences. In Section 6.2, we give a summary of the QPS framework. In Section 6.3 we propose a way to explain qualitative preferences by the deciding criteria, and discuss in particular how this can be implemented for QPS models. In Section 6.4 we discuss how such explanations, if given by the user of a system, can be used to update the system's current model of the user's preferences. We give detailed interaction diagrams that indicate when and how a QPS preference model should be altered. Section 6.5 concludes the paper.

6.2 Qualitative Preference Systems

The main aim of the Qualitative Preference System (QPS) framework [121, 130] is to determine preferences between *outcomes* in a purely *qualitative* way. Outcomes are defined as variable assignments that respect the constraints in a *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*. We distinguish between simple and compound criteria. Simple criteria are based on a single variable. Multiple (simple) criteria can be combined in a compound criterion to determine an overall preference. QPS distinguishes between two kinds of compound criteria: cardinality criteria and lexicographic criteria. The subcriteria of a cardinality criterion all have equal priority, and preference is determined by a kind of voting mechanism that counts the number of subcriteria that support a certain preference and those that do not. In a lexicographic criterion, the subcriteria are ordered by priority and preference is determined by the subcriteria with the highest priority; lower priority subcriteria only influence the preference if the higher priority subcriteria are indifferent.

Definition 6.1. (Qualitative Preference System [121]) A *Qualitative Preference System (QPS)* is a tuple $\langle Var, Dom, K, \mathcal{C} \rangle$. *Var* is a finite set of *variables*. Every variable $X \in Var$ has a domain $Dom(X)$ of possible values. K (a *knowledge base*) is a set of constraints on the assignments of values to the variables in *Var*. 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. Ω denotes the set of all outcomes: $\Omega \subseteq \prod_{X \in Var} Dom(X)$. α_X denotes the value of variable X in outcome α. $\mathcal{C}$ is a finite rooted tree of criteria, where leaf nodes are simple criteria and other nodes are compound criteria. Child nodes of a compound criterion are called its subcriteria. The root of the tree is called the top criterion. 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. $\alpha \curlywedge_c \beta$ denotes that $\alpha \not\succeq_c \beta$ and $\beta \not\succeq_c \alpha$.

Definition 6.2. (Simple criterion [121]) A *simple criterion* c is a tuple $\langle X_c, \trianglerighteq_c \rangle$, where $X_c \in Var$ is a variable, and $\trianglerighteq_c$, a preference relation on the possible values of X_c, is a preorder on $Dom(X_c)$. $\vartriangleright_c$ is the strict subrelation, $\doteq_c$ is the indifference subrelation. A simple criterion $c = \langle X_c, \trianglerighteq_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $\alpha_{X_c} \trianglerighteq_c \beta_{X_c}$.

Definition 6.3. (Goal [130]) A QPS *goal* is a simple criterion $\langle X, \trianglerighteq \rangle$, where $X \in Var$ is a Boolean variable ($Dom(X) = \{\top, \bot\}$), and $\top \vartriangleright \bot$.

Definition 6.4. (Goal-based cardinality criterion [130]) A *goal-based cardinality criterion* c is a tuple $\langle C_c \rangle$ where C_c is a nonempty set of goals (the *subcriteria* or *subgoals* of c). A goal-based cardinality criterion $c = \langle C_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $|\{s \in C_c \mid \alpha \succ_s \beta\}| \geq |\{s \in C_c \mid \alpha \not\succeq_s \beta\}|$, or equivalently, iff $|\{s \in C_c \mid \alpha_{X_s} = \top\}| \geq |\{s \in C_c \mid \beta_{X_s} = \top\}|$.

Note that a goal-based cardinality criterion can only have goals as subcriteria. This is to guarantee transitivity of the preference relation induced by a cardinality criterion [121].

Definition 6.5. (Lexicographic criterion [121]) A *lexicographic criterion* c is a tuple $\langle C_c, \vartriangleright_c \rangle$, where C_c is a nonempty set of criteria (the *subcriteria* of c) and $\vartriangleright_c$, a *priority relation* among subcriteria, is a strict partial order (a transitive and asymmetric relation) on C_c. $s \vartriangle_c s'$ denotes that $s \ntriangleright_c s'$ and $s' \ntriangleright_c s$. A lexicographic criterion $c = \langle C_c, \vartriangleright_c \rangle$ *weakly prefers* an outcome α over an outcome β, denoted $\alpha \succeq_c \beta$, iff $\forall s \in C_c(\alpha \succeq_s \beta \vee \exists s' \in C_c(\alpha \succ_{s'} \beta \wedge s' \vartriangleright_c s))$.

6.3 Explaining preferences

Ideally, any explanation given to a human user should be easily understandable by that user. Therefore, both the content and the format of the explanation matter. Labreuche [82] distinguishes between two steps in explanation generation. First, the content of the explanation has to be selected. Next, a natural language explanation has to be generated. Like Labreuche, we focus on the first step and only look at the content of an explanation. An example of natural language generation for evaluative arguments such as explanations can be found in [40].

We are not aware of any work on the explanation of preferences represented in a qualitative framework, but some work has been done on the explanation of (decisions based on) quantitative preferences. Klein and Shortliffe [80] presented strategies for automatically explaining decisions based on Multiattribute Value Theory (a quantitative preference representation framework). The explanations are based on the compellingness of objectives. Labreuche [82] presents a general framework for explaining the results of a multi-attribute preference model. He takes a quantitative approach where the utilities of the combined criteria are weighted and summed to obtain an overall utility. He develops a formal framework that justifies the selection of arguments (criteria) to be presented as explanation of a preference.

One of the main differences between quantitative and qualitative approaches to multi-criteria preference modelling is that quantitative approaches are compensatory, whereas their qualitative counterparts are not. In quantitative approaches, a low score on one criterion can be compensated by high scores on other criteria, even if the other criteria are less important, as long as the scores are high enough. In qualitative approaches, this is not possible. For example, if one outcome is preferred to another according to the highest priority subcriterion of a lexicographic criterion, it will also be preferred according to this lexicographic criterion, no matter what the preferences of the other subcriteria are. This allows us to precisely identify the criteria that are 'responsible' or 'deciding' for the overall preference. It is our intuition that these criteria also provide a natural explanation for the overall preference.

Explanations for preferences by QPS criteria

We now turn to the question how a preference between two outcomes by a QPS criterion can be explained. The answer to this question depends on the kind of criterion that is considered. Preferences by simple criteria (including goals) are self-explanatory, since they follow immediately from the specification of the simple criterion or goal. For example, a simple criterion c strictly prefers an outcome α to an outcome β because α's value of X_c is better than β's value of X_c. Similarly, a goal c strictly prefers an outcome α to an outcome β because α satisfies c but β does not. Of course, these facts may in turn require explanation. But since this would be explanation of knowledge (factual information about outcomes) rather than preferences, we do not discuss this topic here.

Preferences by compound criteria can be explained by the subcriteria that are *deciding* in the overall preference. Which subcriteria are deciding depends both on the kind of compound criterion (lexicographic or goal-based cardinality criterion) and on the kind of preference (strict, equal or incomparable). The deciding factor may be a single subcriterion, a pair, or even a set of multiple subcriteria that together determine the overall preference. In the following, we discuss the deciding subcriteria (and hence the explanations) for both kinds of compound criteria and for all kinds of preferences. An overview is given in Table 6.1.

	lexicographic criterion c	goal-based cardinality criterion c
$\alpha \succ_c \beta$	any subcriterion $s \in C_c$ such that $\alpha \succ_s \beta$ and for all $s' \in C_c$: if $s' \rhd s$ then $\alpha \approx_{s'} \beta$ and if $s' \triangle s$ then $\alpha \succeq_{s'} \beta$ or there is a $s'' \in C_c(s'' \rhd_c s'$ and $\alpha \not\approx_{s''} \beta)$	the set of subgoals $g \in C_c$ such that $\alpha \succ_g \beta$
$\alpha \approx_c \beta$	for all subcriteria $s \in C_c$: $\alpha \approx_s \beta$	the set of subgoals $g \in C_c$ such that $\alpha \succ_g \beta$ plus the set of subgoals $g \in C_c$ such that $\beta \succ_g \alpha$
$\alpha \curlywedge_c \beta$	1: any subcriterion $s \in C_c$ such that $\alpha \curlywedge_s \beta$ and for all $s' \in C_c$: if $s' \rhd s$ then $\alpha \approx_{s'} \beta$ 2: any pair of subcriteria (s_1, s_2) where $s_1, s_2 \in C_c$ such that $\alpha \succ_{s_1} \beta$ and $\beta \succ_{s_2} \alpha$ and $s_1 \triangle_c s_2$ and for all $s' \in C_c$: if $s' \rhd_c s_1$ or $s' \rhd_c s_2$ then $\alpha \approx_{s'} \beta$	n/a

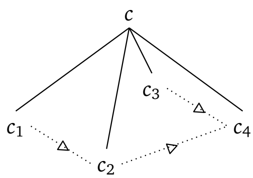
Table 6.1: Explanations

Lexicographic criteria

Strict preference Suppose a lexicographic criterion c strictly prefers an outcome α over an outcome β ($\alpha \succ_c \beta$). The explanation of this preference is given by a subcriterion s that strictly prefers α to β ($\alpha \succ_s \beta$). But not just any subcriterion that strictly prefers α to β will do. First, every subcriterion s' with a higher priority than s ($s' \rhd_c s$) has to be indifferent: $\alpha \approx_{s'} \beta$, otherwise s would have been overruled by s'. Second, every subcriterion s' whose priority is incomparable to that of s ($s' \triangle_c s$) and which is not overruled ($\forall s'' \rhd_c s' : \alpha \approx_{s''} \beta$) has to agree with s or be indifferent ($\alpha \succeq_{s'} \beta$), otherwise s would not have decided the preference by c.

Example 6.1. Consider the lexicographic criterion c displayed in Figure 6.1. It has four subcriteria c_1, c_2, c_3, c_4 such that $c_1 \rhd_c c_2 \rhd_c c_4$ and $c_3 \rhd_c c_4$. The nature of the subcriteria is unspecified, but their preferences regarding four outcomes $\alpha, \beta, \gamma, \delta$ are given. The criterion c strictly prefers α over β: $\alpha \succ_c \beta$. The subcriteria that can explain this preference are c_2 and c_3. c_3 strictly prefers α over β, and is undominated. c_2 also strictly prefers α over β, and is dominated only by an indifferent criterion (c_1). Neither is 'contradicted' by a criterion with incomparable priority.

Equal preference A lexicographic criterion c is only indifferent between two outcomes α and β ($\alpha \approx_c \beta$) if all its subcriteria are indifferent between α and β. No single subcriterion is deciding in the overall preference, but all subcriteria contribute equally (note that priority does not matter, since indifferent criteria do not overrule



$$\alpha \approx_{c_1} \beta \approx_{c_1} \gamma >_{c_1} \delta$$
$$\alpha >_{c_2} \beta \approx_{c_2} \gamma >_{c_2} \delta$$
$$\alpha, \delta >_{c_3} \beta \approx_{c_3} \gamma;\ \alpha \curlywedge_{c_3} \delta$$
$$\gamma \approx_{c_4} \beta >_{c_4} \delta >_{c_4} \alpha$$

Figure 6.1: Example lexicographic criterion

lower priority criteria). This means that the explanation of the indifference is given by the fact that all subcriteria are indifferent.

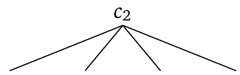
Example 6.2. Consider again the lexicographic criterion c in Figure 6.1. c is indifferent between β and γ, because all subcriteria are indifferent between β and γ.

Incomparability If a lexicographic criterion c cannot compare between two outcomes α and β ($\alpha \curlywedge_c \beta$), this incomparability can have two possible reasons. First, the incomparability may result from a subcriterion s that cannot compare between α and β ($\alpha \curlywedge_s \beta$). Like in the case of strict preference, every subcriterion s' with a higher priority than s ($s' \rhd_c s$) has to be indifferent: $\alpha \approx_{s'} \beta$, otherwise s would have been overruled by s'.

Example 6.3. Consider again the lexicographic criterion c in Figure 6.1. c cannot compare between α and δ. This is due to subcriterion c_3, which cannot compare between α and δ, and which is not overruled by any other subcriterion. Therefore c_3 explains c's incomparability between α and δ.

Second, the incomparability may result from two conflicting subcriteria that do not overrule each other. That is, there is one subcriterion s_1 that strictly prefers α to β ($\alpha >_{s_1} \beta$), and all higher priority subcriteria are indifferent. There is also another subcriterion s_2 that strictly prefers β to α ($\beta >_{s_2} \alpha$), and all higher priority subcriteria are indifferent. Note that this also means that s_1 and s_2 have incomparable priorities, which means that neither overrules the other, so no preference can be determined. In this case, the subcriteria s_1 and s_2 together explain the incomparability.

Example 6.4. Consider again the lexicographic criterion c in Figure 6.1. c cannot compare between γ and δ. Subcriterion c_3 strictly prefers δ over γ, the other three subcriteria stricly prefer γ over δ. Not all subcriteria are suitable to explain the incomparability. c_4 is discarded because c_3 has higher priority. But also c_2 should not be used, even though it is incomparable in priority with c_3. This is because c_1 has higher priority and is not indifferent. This makes c_1 and c_3 the deciding criteria that are used as explanation.



	g_1	g_2	g_3	g_4
α	$\top$	$\top$	$\top$	$\bot$
β	$\bot$	$\bot$	$\top$	$\top$
γ	$\top$	$\bot$	$\bot$	$\top$

Figure 6.2: Example goal-based cardinality criterion

Goal-based cardinality criteria

Strict preference Suppose a goal-based cardinality criterion c strictly prefers an outcome α over an outcome β ($\alpha \succ_c \beta$). Then this is because the subgoals that α satisfies outnumber the subgoals that β satisfies. There may be subgoals that are satisfied by both α and β. They are counted on both sides, but do not influence the overall preference between α and β. Therefore, as an explanation of c's preference of α over β we only consider the subgoals g that α satisfies but β does not ($\alpha_{X_g} = \top$ and $\beta_{X_g} = \bot$, or equivalently, $\alpha \succ_g \beta$).

Example 6.5. Consider the goal-based cardinality criterion c_2 displayed in Figure 6.2. It has four goals g_1, g_2, g_3, g_4 as subcriteria. For three outcomes α, β, γ it is given whether they satisfy each of the four goals. The criterion c_2 strictly prefers α to β. The explanation of this preference is given by the goals g_1 and g_2 that α satisfies but β does not. Although α also satisfies goal g_3, this goal is not used in the explanation since it is also satisfied by β and hence is not deciding in the overall preference. Similarly, c_2's preference of α over γ can be explained by the goals g_2 and g_3.

Equal preference If a goal-based cardinality criterion c equally prefers two outcomes α and β ($\alpha \approx_c \beta$), this means that both outcomes satisfy the same number of subgoals of c. However, it does not necessarily mean that both outcomes satisfy the same goals. As explanation, we take the goals that α satisfies but β does not, and the set of goals that β satisfies but α does not. Both (disjoint) sets contain the same number of goals, which compensate for each other. This explains the indifference between the two outcomes.

Example 6.6. Consider again the goal-based cardinality criterion c_2 in Figure 6.2. c_2 is indifferent between β and γ. Both outcomes satisfy two goals, but one goal (g_4) is satisfied by both outcomes. Therefore the explanation of the indifference is given by g_3 (which is satisfied by β but not by γ) and g_1 (which is satisfied by γ but not by β).

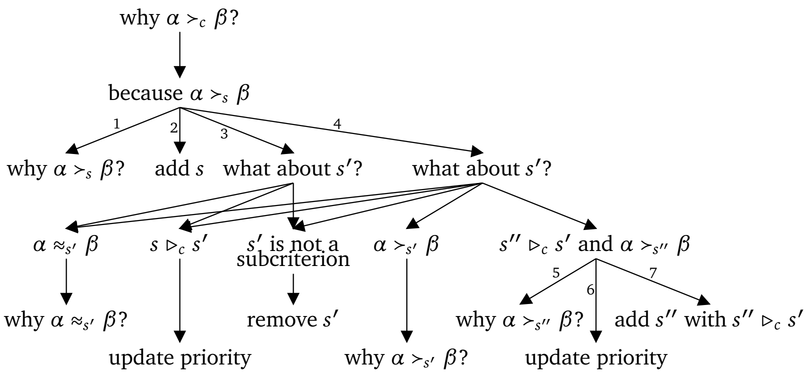
6.4 Using explanation to update a preference model

Before a preference model can be used in practice in a system, it has to be constructed or instantiated. Preference elicitation is likely to be an iterative process, and for this reason an existing preference model should also be updateable. There are several ways of constructing and updating a preference model. In this paper we focus on the approach of guiding preference elicitation by asking the user particular questions and updating the preference model according to the answers. The advantages of this approach are that it provides an intuitive interaction with non-expert users and that preferences can be discovered during the process. In particular, we consider the case in which the user is asked not only to give his preference between two outcomes, but also to provide an explanation for this preference. This explanation can then be used to update the current preference model. If the user just provides his preference between outcomes, there may be many different ways in which the model could be updated to reflect this preference. The added value of additionally obtaining an explanation from the user is that it provides clues on how exactly the model should be updated, possibly after some further interaction involving targeted follow-up questions.

Updating a QPS model with explanations

We investigate how a system's current model of the user's preferences can be updated by engaging in a conversation with the user. Using explanations of preferences given by a user, the system can find out whether its current representation is accurate, and if not, where it has to be changed. Our approach allows for an initial model to be present that can be adapted by the user. The user can add preference information on his own initiative, or alternatively the system can ask the user to provide specific preferences (for example between two outcomes that are incomparable in its current model). In any case, if the preference given by the user does not match the preference that follows from the system's current model, the user is asked to provide an explanation. We assume that the user's explanation of his preference coincides with one of the explanations listed in Table 6.1. Depending on the user's answer and the nature of the top criterion (lexicographic or goal-based cardinality), the system can proceed by asking follow-up questions or updating its preference model in a particular way.

In the following, we discuss every situation in detail and provide interaction diagrams for each. We assume that the user has stated a preference between two outcomes that is not supported by the system's current preference model. It is important to distinguish between the current preference model maintained by the system, and the statements of the user. Since the interaction is designed to identify the elements of the model that need to be updated, the user's statements typically disagree with the current model. The interaction diagrams start with the system asking for an explanation for the given preference. The system's possible responses depend on the explanation given and the current preference model. More than one response may be applicable. In that case, the system should keep the interaction going until



1: if $\alpha \not\succ_s \beta$
2: if $s \notin C_c$

3: for any $s' \in C_c$ such that $s' \rhd_c s$ and $\alpha \not\approx_{s'} \beta$

4: for any $s' \in C_c$ such that $s' \triangle s$ and $\alpha \not\succsim_{s'} \beta$ and s' is not overruled

5: if $\alpha \not\succ_{s''} \beta$
6: if $s'' \not\rhd_c s$
7: if $s'' \notin C_c$

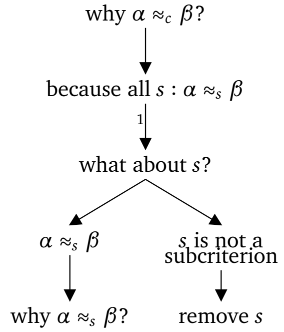
Figure 6.3: Updating with a strict preference by a lexicographic criterion

the preference model induces the given preference. When the process is finished, the updated preference model should not only model the preference given by the user, but also generate the same explanation for it.

Lexicographic criteria

Strict preference The interaction diagram for updating a preference model with a strict preference of an outcome α over an outcome β by a lexicographic criterion c is given in Figure 6.3. The explanation of such a preference is given by a subcriterion s of c that, according to the user, strictly prefers α to β. There can be different reasons why this subcriterion does not decide c's preference in the current preference model S.

- First, s may not strictly prefer α to β according to S. In this case, the user is asked to explain this preference.
- Second, s may not be listed as a subcriterion of c in S. In this case, the system adds s to the set of subcriteria C_c.
- Third, according to S there may be another subcriterion s' that overrules s, i.e. that has higher priority but is not indifferent between α and β. In this case, the user is asked to clarify this issue, and may respond in several ways. (i) If the user states that s' actually is indifferent, he is asked for an explanation. (ii) If the user states that s actually has higher priority than s', the system updates the priority relation accordingly. (iii) If the user states that s' is not actually a subcriterion, the system removes s' from C_c.
- Fourth, according to S there may be another subcriterion s' that is not comparable in priority to s, does not weakly prefer α to β, and is not overruled. In this case,



1: for any $s \in C_c$ such that $\alpha \not\approx_s \beta$

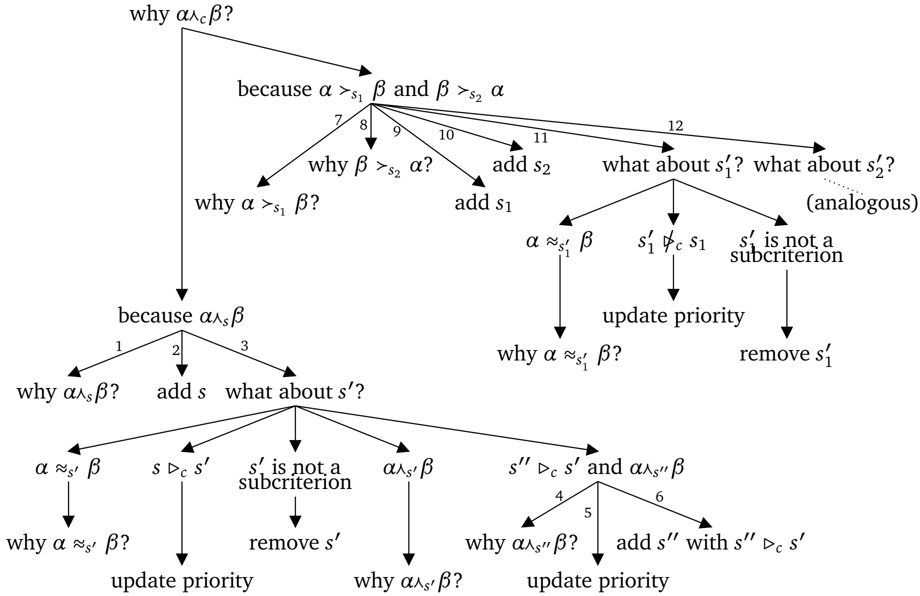
Figure 6.4: Updating with an equal preference by a lexicographic criterion

the user is asked to clarify this issue. The same responses by the user as in the previous case are possible, plus two more. (iv) If the user states that s' actually strictly prefers α to β, he is asked to give an explanation. (v) If the user states that there actually is another subcriterion s'' with higher priority that strictly prefers α to β, there are three options. If the preference does not follow from S, then the user is asked for an explanation. If s'' does not have higher priority than s' in S, the system updates the priority relation. And if s'' was not listed as a subcriterion of c, the system adds it with the right priority.

Equal preference The interaction diagram for updating a preference model with an equal preference between two outcomes α and β by a lexicographic criterion c is given in Figure 6.4. Such a preference is explained by the fact that, according to the user, all subcriteria are indifferent. There can only be one reason that the indifference does not follow from the current preference model S.

- There must be a subcriterion s in S that is not indifferent. In this case, the user is asked to clarify this issue. He can do so in two different ways. (i) If the user states that s is actually indifferent, he is asked to give an explanation. (ii) If the user states that s is not actually a subcriterion of c, then the system removes s from the set of subcriteria C_c.

Incomparability The interaction diagram for updating a preference model with an incomparability between two outcomes α and β by a lexicographic criterion c is given in Figure 6.5. Since there are two kinds of explanation of such an incomparability, the interaction tree splits into two branches. If the incomparability is explained by a subcriterion that cannot compare between α and β according to the user, the possible responses are very similar to the case of strict preference. Therefore we do not discuss this case here but refer to the lefthand branch in Figure 6.5 for the details. If the



1: if $\alpha \not\curlywedge_s \beta$
2: if $s \notin C_c$
3: for any $s' \in C_c$ such that $s' \triangleright_c s$ and $\alpha \succ_{s'} \beta$ or $\beta \succ_{s'} \alpha$

4: if $\alpha \not\curlywedge_{s''} \beta$
5: if $s'' \not\triangleright_c s'$
6: if $s'' \notin C_c$

7: if $\alpha \not\succ_{s_1} \beta$
8: if $\beta \not\succ_{s_2} \alpha$
9: if $s_1 \notin C_c$
10: if $s_2 \notin C_c$

11: for any $s_1' \in C_c$ such that $s_1' \triangleright_c s_1$ and $\alpha \not\approx_{s_1'} \beta$
12: for any $s_2' \in C_c$ such that $s_2' \triangleright_c s_2$ and $\alpha \not\approx_{s_2'} \beta$

Figure 6.5: Updating with an incomparability by a lexicographic criterion

incomparability is explained by two contradicting subcriteria s_1 and s_2, where $\alpha \succ_{s_1} \beta$ and $\beta \succ_{s_2} \alpha$ according to the user, there can be different reasons why these subcriteria do not decide c's preference in the current preference model S.

- First, it may be that $\alpha \not\succ_{s_1} \beta$ or $\beta \not\succ_{s_2} \alpha$ according to the current preference model S. In this case, the user is asked to explain that preference.
- Second, s_1 or s_2 may not be listed as a subcriterion of c in S. In this case, the system adds it to the set of subcriteria C_c.
- Third, according to S there may be another subcriterion s_1' that overrules s_1. In this case, the user can reply in different ways. (i) If the user states that s_1' is actually indifferent between α and β, he is asked for an explanation. (ii) If the user states that s_1' does not actually have higher priority than s_1, the system updates the priority relation accordingly. (iii) If the user states that s_1' is not actually a subcriterion of c, then the system removes s_1' from the set of subcriteria C_c.
- Fourth, according to S there may be another subcriterion s_2' that overrules s_2. This case is handled analogously to the third case.

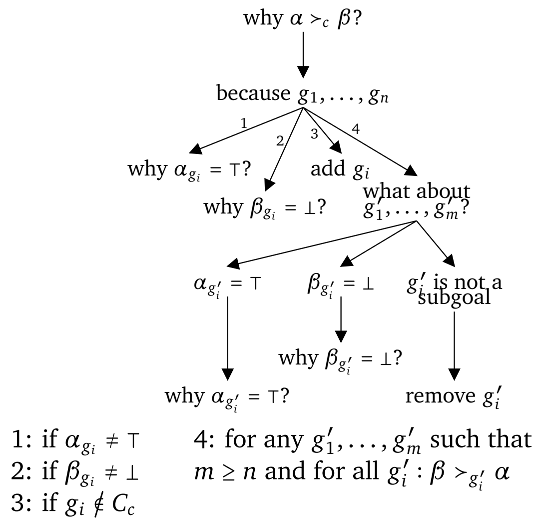


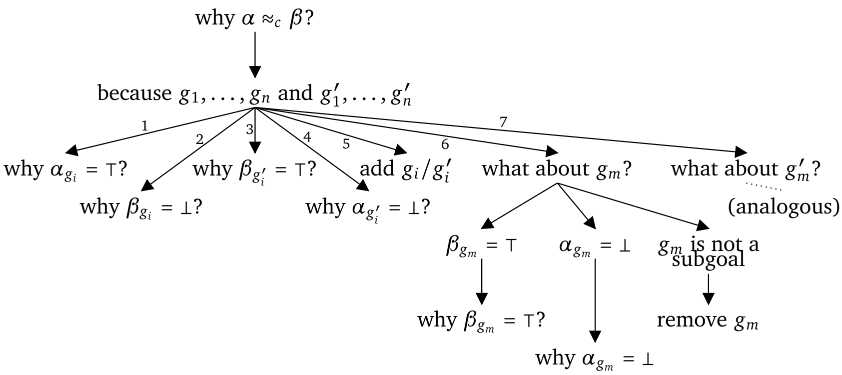
Figure 6.6: Updating with a strict preference by a goal-based cardinality criterion

Goal-based cardinality criteria

Strict preference The interaction diagram for updating a preference model with a strict preference of an outcome α over an outcome β by a goal-based cardinality criterion c is given in Figure 6.6. The explanation of such a preference is given by a set of subgoals $g_1, \ldots, g_n$ that are all satisfied by α but not by β according to the user. There can be different reasons why this set of goals does not decide c's preference in the current preference model S.

- First, one of the goals may not be satisfied by α in S. In this case, the user is asked to explain this fact.
- Second, one of the goals may be satisfied by β in S. In this case, the user is also asked to give an explanation.
- Third, one of the goals may not be listed as a subgoal of c in S. In this case, the system adds it to the set of subgoals C_c.
- Fourth, there may be a set of goals $g'_1, \ldots, g'_m$ that are all satisfied by β but not by α according to S, which contains at least as many goals as $g_1, \ldots, g_n$. In this case, the user is asked to clarify this issue, and may respond in several ways. (i) If the user states that one of the goals is actually satisfied by α or (ii) not satisfied by β, he is asked to for an explanation. (iii) If the user states that one of the goals is actually not a subgoal of c, then the system removes this goal from the set of subgoals C_c.

Equal preference The interaction diagram for updating a preference model with an equal preference between two outcomes α and β by a goal-based cardinality



1: if $\alpha_{g_i} \neq \top$	4: if $\alpha_{g'_i} \neq \bot$	6: for any $g_m \in C_c$ such	7: for any $g'_m \in C_c$ such
2: if $\beta_{g_i} \neq \bot$	5: if $g_i/g'_i \notin C_c$	that $g_m \notin g_1, \ldots, g_n$	that $g'_m \notin g'_1, \ldots, g'_n$
3: if $\beta_{g'_i} \neq \top$		and $\alpha >_{g_m} \beta$	and $\beta >_{g'_m} \alpha$

Figure 6.7: Updating with an equal preference by a goal-based cardinality criterion

criterion c is given in Figure 6.7. The explanation of such a preference is given by two equally sized sets of subgoals: $g_1, \ldots, g_n$ that are all satisfied by α but not by β, and $g'_1, \ldots, g'_n$ that are all satisfied by β but not by α according to the user. Again, there can be different reasons why these sets of goals do not decide c's preference in the current preference model S.

- First, according to S, α may not satisfy some g_i, β may satisfy some g_i, β may not satisfy some g'_i, or α may satisfy some g'_i. In this case, the user is asked to give an explanation.
- Second, any g_i or g'_i may not be listed as a subgoal of c in S. In this case, the system adds it to the set of subgoals C_c.
- Third, according to S there may be a goal g_m in C_c that is not in $g_1, \ldots, g_n$ and is satisfied by α but not by β. In this case, the user is asked to clarify this issue and may respond in several ways. (i) If the user states that β actually satisfies g_m, or (ii) α actually does not satisfy g_m, he is asked to explain this fact. (iii) If the user states that g_m is actually not a subgoal of c, then the system removes g_m from the set of subgoals C_c.
- Fourth, according to S there may be a goal g'_m in C_c that is not in $g'_1, \ldots, g'_n$ and is satisfied by β but not by α. This case is handled analogously to the third case.

6.5 Conclusion

Qualitative Preference Systems (QPS) [121, 130] provide a general framework for the representation of qualitative, multi-criteria preferences. We have shown that the composite tree structure of multiple criteria, combined with the non-compensatoriness of a qualitative approach provides a basis for the generation of explanations for the preferences that follow from a preference model represented in the QPS framework. The explanation strategy that we proposed is based on the intuition that preferences between outcomes can be explained by the criteria that are deciding in the overall preference. We identified the explanations that can be given for different preferences by different kinds of criteria. We then showed that the same explanations can also be useful when updating a preference model, because they provide information on how exactly the model should be updated.

Some interesting issues remain for future work. First, in some instances it may be necessary to explain facts about the outcomes involved in a preferential comparison, e.g. to explain why they do or do not satisfy a particular goal. Explanation of knowledge and reasoning is a separate field of study that may provide solutions to this issue. Second, when the system updates the priority relation between two subcriteria of a lexicographic criterion, this relation has to remain a partial order. Moreover, as the system iteratively engages in an interaction with the user as described here, it has to ensure that the previous preferences and explanations expressed by the user remain valid. It is important to investigate how such consistencies can be ensured. Third, the explanation of preferences may be part of a larger picture, for example in recommendation, decision making or planning. We would like to investigate how the explanation mechanism presented here can be embedded in other explanation mechanisms, such as the one presented in [38], where a tree structure of goals and beliefs is used to explain actions. Besides these theoretical considerations, we would like to take a more practical approach and implement the QPS framework together with the proposed explanation mechanism and update mechanism. We can then experimentally test the validity of our intuitions. This is related to the work of Dieckmann *et al.* [45], who tested the predictive performance of the Take the Best (TTB) heuristic [57], which is a simplified instantiation of the lexicographic rule.

Chapter 7

Multi-Attribute Preference Logic

Abstract Preferences for objects are commonly derived from ranked sets of properties or multiple attributes associated with these objects. There are several options or strategies to qualitatively derive a preference for one object over another from a property ranking. We introduce a modal logic, called Multi-Attribute Preference Logic, that provides a language for expressing such strategies. The logic provides the means to represent and reason about qualitative multi-attribute preferences and to derive object preferences from property rankings. The main result of the paper is a proof that various well-known preference orderings can be defined in Multi-Attribute Preference Logic.

7.1 Introduction

Preferences may be associated with various entities such as states of affairs, properties, objects and outcomes in e.g. games. Our main concern here are object preferences. A natural 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, where the ranking indicates the relative importance or priority of each of these properties. This approach to obtain preferences is typical in multi-attribute decision theory, see e.g. Keeney and Raiffa [76]. Multi-attribute decision theory provides 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. As it is difficult to obtain such quantitative utility values and weights, however, several qualitative approaches have been proposed instead, see e.g. [28, 33, 34, 44, 86]. There is also extensive literature on preference logic following the seminal work of Von Wright [138, 61], but such logics are not specifically suited for the multi-attribute case. To illustrate what we are after, we first present a

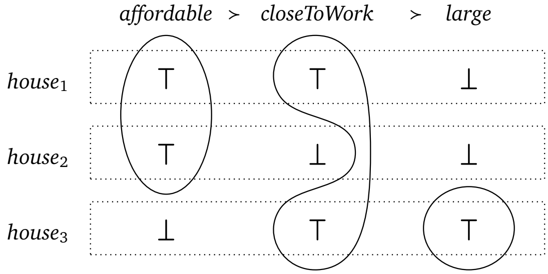


Figure 7.1: Properties of three houses

motivating example that is used throughout the paper.

Example 7.1. Suppose we want to buy a house. The properties that we find important are that we can afford the house, that it is close to our work, and that it is large, in that order. Consider three houses, $house_1$, $house_2$ and $house_3$, whose properties are listed in Figure 7.1, which we have to order according to our preferences. It seems clear that we would prefer $house_1$ over the other two, because it has two of the most important properties, while both other houses only have one of these properties. But what about the relative preference of $house_2$ and $house_3$? $house_3$ has two out of three of the relevant properties where $house_2$ has only one. If the property that $house_2$ has is considered more important than both properties of $house_3$, $house_2$ would be preferred over $house_3$.

Key to a logic of multi-attribute preferences is the representation of property rankings. Encodings of property rankings have been explored by Coste-Marquis *et al.* [44] where they are called goal bases, and by Brewka [33] where they are called ranked knowledge bases. Such ranked goals are binary, and in this paper we also consider desired attributes that are binary (as opposed to numeric or ordinal ones). Coste-Marquis *et al.* and Brewka moreover discuss various options, or strategies, for deriving object preferences from a property ranking. The preference orderings thus obtained are not expressed in a logic, however. Brewka *et al.* [34] propose a non-monotonic logic called qualitative choice logic to reason about multi-attribute preferences. An alternative approach towards a logic of multi-attribute preferences is presented in Liu [86] where property rankings called priority sequences are encoded in first-order logic. Both approaches are based on one particular strategy, namely lexicographic ordering, and cannot be used to reason about preference orderings.

In this paper a generic logic of qualitative multi-attribute preferences is proposed in which property rankings and associated strategies for deriving object preferences from such rankings can be defined. In Section 7.2 the syntax and semantics of Multi-Attribute Preference Logic is introduced. Section 7.3 shows how various strategies to obtain object preferences from a property ranking can be defined in the logic. Section 7.4 presents the main result of the paper and shows that property rankings encoded as ranked knowledge bases and a number of related strategies to obtain preference orderings can be equivalently translated into Multi-Attribute Preference Logic. Section 7.5 concludes the paper.

7.2 Multi-Attribute Preference Logic

7.2.1 Syntax and semantics

The logic of multi-attribute preferences that we introduce here is an extension of the modal binary preference logic presented by Girard [58]. This logic is a propositional modal logic with a modal operator $\Box^{\leq}\varphi$, and its dual $\Diamond^{\leq}\varphi$. Here $\Box^{\leq}\varphi$ expresses that φ is true in all states that are at least as good as the current state. Binary preference relations over formulae are subsequently defined. One of the more natural binary preference statements is $\varphi <_{\forall\forall} \psi$ which expresses that any state where ψ is true is strictly better than any state where φ is true. That is, whenever φ is the case, ψ is preferred, and never vice versa. By adding a global modality U to the language, the binary preference operator $<_{\forall\forall}$ can be defined by $U(\psi \to \Box^{\leq}\neg\varphi)$, when it is assumed that the underlying order on worlds or states has been completely specified, i.e. is total.

Multi-Attribute Preference Logic adds two operators to binary preference logic. First, Multi-Attribute Preference Logic, as in hybrid logic [23] adds names for objects to the language by adding nullary modal operators i, j to the language. The semantics of the operators introduced here, however, differs from the standard semantics of hybrid logic. Here i, j are used as names for objects which semantically are more complex entities than the usual worlds of modal semantics. In order to avoid confusion, we will refer to i, j as object names below. This language extension allows us to talk about objects and associated preferences explicitly.

Second, the logic introduces a new modal operator $\Box^{\neq}$. The language of Multi-Attribute Preference Logic consists of four unary modal operators. Instead of the single operator $\Box^{\leq}$ it is more convenient to introduce the two operators $\Box^{<}$ and $\Box^{=}$: informally, $\Box^{<}\varphi$ expresses that at all worlds that are ranked higher than the current one φ is true, whereas $\Box^{=}\varphi$ expresses that at all worlds that are equally ranked to the current one φ is true. The modal operator $\Box^{\neq}$ is introduced to inspect worlds that are not ranked equally to the current one.

Definition 7.1. (Language) Let At be a set of propositional atoms with typical element p and Nom be a set of names, with typical elements i, j. The language $\mathcal{L}_{pref}$ is defined as follows:

$$\varphi \in \mathcal{L}_{pref} \quad ::= \quad p \mid i \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box^{=}\varphi \mid \Box^{\neq}\varphi \mid \Box^{<}\varphi \mid U\varphi$$

Disjunction $\vee$, implication $\to$, and bi-implication $\leftrightarrow$ are defined as the usual abbreviations. $\Diamond^{<}\varphi, \Diamond^{=}, \Diamond^{\neq}$ are abbreviations for $\neg\Box^{<}\neg\varphi$, $\neg\Box^{=}\neg\varphi$, and $\neg\Box^{\neq}\neg\varphi$. $\Box^{\leq}\varphi$ is short for $\Box^{<}\varphi \wedge \Box^{=}\varphi$ and $\Diamond^{\leq}\varphi$ is its dual. The dual of the global modal operator,

$E\varphi$, is defined as $\neg U \neg \varphi$. We also write $U_i\varphi$ for $U(i \to \varphi)$ and $E_i\varphi$ for $E(i \wedge \varphi)$ for $i \in Nom$. Finally, the set of purely propositional formulae is denoted by $\mathcal{L}_0$ and consists of all formulae without any occurrences of modal operators or names $i \in Nom$. $\varphi \in \mathcal{L}_0$ is also called an objective formula.

The basic concepts in the semantics for Multi-Attribute Preference Logic are objects and properties those objects may have. Properties are naturally represented by sets of worlds. As we want to use properties to classify the ranking of objects, properties are ordered in correspondence with their relative importance; such an order is called a property ranking here. To order properties, i.e. sets of worlds, it is required that properties are disjoint sets of worlds. Property rankings will be derived from an order on worlds below.

Objects are also identified with particular sets of worlds. The idea is that the properties (in the sense of the previous paragraph) of an object can be derived from the worlds which define the object. To ensure that objects are coherent, that is have a uniquely defined set of properties, the worlds that define the object need to be copies of each other, which means that these worlds need to assign the same truth values to propositional atoms. Objects are identified with equivalence classes of worlds with respect to a truth assignment.

Definition 7.2. (Object) Let W be a set of worlds and V be a mapping of W to truth assignments 2^{At}. An *object* is an equivalence class on W with respect to V. The set $\mathcal{O}_V$ denotes the set of all objects defined by W and V and is formally defined by:

$$\mathcal{O}_V = \{[w]_V \mid w \in W\}$$

where $[w]_V = \{v \in W \mid V(w) = V(v)\}$. Whenever V is clear from the context, we drop the subscript V. As an object o is the equivalence class of a world w with respect to V, we also say that world w *identifies* object o.

Definition 7.3. (Model) A *multi-attribute preference model* $\mathcal{M}$ is a tuple $\langle W, \precsim, V, N\rangle$ where W is a set of *worlds* with typical elements u, v, w, $\precsim$ is a *total pre-order* (i.e. a reflexive, transitive and total relation) on W, V is a *valuation function* mapping worlds in W onto truth assignments in 2^{At}, and N is a *naming function*. The strict subrelation $\prec$ of $\precsim$ is defined by: $v \prec w := v \precsim w \ \& \ w \not\precsim v$. We write $v \sim w$ whenever $v \precsim w$ and $w \precsim v$.

Although the strict order $\prec$ derived from $\precsim$ indicates a ranking of worlds where $v \prec w$ means that w is ranked higher than v, we do not say that w is preferred over v, because we want to reserve this terminology for talking about objects. A preference between objects is derived from the ranking $\precsim$ over worlds. The naming function N maps names i to objects o.

The truth definition for propositional atoms and Boolean operators is standard. Given a model $\mathcal{M} = \langle W, \precsim, V, N\rangle$, the semantics of names $i \in Nom$ is provided by the naming function N. The truth definitions for most modal operators are also standard definitions using the associated accessibility relations for these operators.

The semantic clause for $\Box^{=}$ is defined by means of the relation $\sim$, which is derived from the order $\lesssim$. Similarly, the semantic clause for $\Box^{<}$ is provided by means of the strict order $<$. The global operator U simply inspects all worlds in a model.

The truth definition for $\Box^{\neq}$ is not directly defined in terms of a given relation on W. It inspects all worlds that (i) are not ranked equally as the current one, and (ii) are not copies of worlds that are ranked equally as the current one. The motivation for this definition will become clear in Section 7.2.2 when clusters are introduced.

Definition 7.4. (Truth Definition) Let $\mathcal{M} = \langle W, \lesssim, V, N\rangle$ be an MPL model and $w \in W$ a world. The truth of a formula $\varphi \in \mathcal{L}_{pref}$ in $\mathcal{M}$ at w is defined by:

$$
\begin{array}{lcl}
\mathcal{M}, w \vDash p & \Leftrightarrow & p \in V(w) \\
\mathcal{M}, w \vDash i & \Leftrightarrow & w \in N(i) \\
\mathcal{M}, w \vDash \neg\varphi & \Leftrightarrow & \mathcal{M}, w \nvDash \varphi \\
\mathcal{M}, w \vDash \varphi \wedge \psi & \Leftrightarrow & \mathcal{M}, w \vDash \varphi \;\&\; \mathcal{M}, w \vDash \psi \\
\mathcal{M}, w \vDash \Box^{=}\varphi & \Leftrightarrow & \forall v : w \sim v \;\Rightarrow\; \mathcal{M}, v \vDash \varphi \\
\mathcal{M}, w \vDash \Box^{\neq}\varphi & \Leftrightarrow & \forall u \in \overline{\bigcup\{[v]_V \mid w \sim v\}} : \mathcal{M}, u \vDash \varphi \\
\mathcal{M}, w \vDash \Box^{<}\varphi & \Leftrightarrow & \forall v : w < v \;\Rightarrow\; \mathcal{M}, v \vDash \varphi \\
\mathcal{M}, w \vDash U\varphi & \Leftrightarrow & \forall v : \mathcal{M}, v \vDash \varphi
\end{array}
$$

A name $i \in Nom$ refers to an object o and, semantically, is true at a world w that identifies the object o, i.e. $w \in o$. A name thus is a special kind of operator that is true in all worlds that identify a certain object, and false in all other worlds. We can express that an object i has a property φ by $E_i\varphi = E(i \wedge \varphi)$. As we have $E(i)$ as a validity and the worlds that identify the corresponding object o are copies of each other, we have $E_i\varphi \leftrightarrow U_i\varphi$ for objective φ. This shows that an object is coherent in the sense that an object has a consistent set of objective properties and can be uniquely identified by this set.

The language also allows us to express properties that concern comparison of objects. For example, $U(i \to \Diamond^{<} j)$ expresses that for every property of object i object j has a property that is strictly better. The formula $E(j \wedge \neg \Diamond^{\leq} i)$ expresses that object j has a property that object i cannot match, i.e. i has no property that is strictly better than this property of j. We have $E(j \wedge \neg \Diamond^{\leq} i) \to U(i \to \Diamond^{<} j)$ in Multi-Attribute Preference Logic. This validity is based on the assumption that the pre-order in models for $\mathcal{L}_{pref}$ is total.

Recall that the binary preference operator $\varphi <_{\forall\forall} \psi$ can be defined as $U(\psi \to \Box^{\leq}\neg\varphi)$. Using $<_{\forall\forall}$ it is possible to define property rankings and express that a property ψ is ranked higher than property φ. Using the truth definitions for $U\varphi$, $\Box^{=}\varphi$ and $\Box^{<}\varphi$ and the definition of $\Box^{\leq}\varphi$ as $\Box^{=}\varphi \wedge \Box^{<}\varphi$, it can be shown that $\varphi <_{\forall\forall} \psi$ has the following truth definition:

$$\mathcal{M}, w \vDash \varphi <_{\forall\forall} \psi \Leftrightarrow \forall u, v : \mathcal{M}, u \vDash \varphi \;\&\; \mathcal{M}, v \vDash \psi \Rightarrow u < v$$

The intuitive reading of $\varphi <_{\forall\forall} \psi$ is that every ψ-state is ranked higher than every φ-state (cf. [58]). Returning to the comparison of objects again, $i <_{\forall\forall} j$ expresses that object j is preferred over i. The preference expressed in this way is a very strong kind of preference, however. It requires that all of object j's relevant properties are considered more important than objects i's properties, which corresponds with the definition of $i <_{\forall\forall} j$ by $U(j \to \Box^{\leq}\neg i)$. In contrast, Multi-Attribute Preference Logic is able to specify principles that allow to derive preferences over objects from their properties in a weaker sense. It enables, for example, to specify orderings where object j is preferred over object i even when object i has at least one property that is considered more important than a property that object j has (compare e.g. object c and f in Figure 7.2). The logic thus facilitates the specification of different ordering strategies, and, given such a specification, provides the means to derive a preference of one object over another from a property ranking and an additional specification of the objects' properties.

Proposition 7.1 supports our claim that multi-attribute preference logic extends binary preference logic as all listed axioms of this logic are valid in Multi-Attribute Preference Logic as well (cf. [58], p. 66). We have listed only those axioms that can straightforwardly be expressed without the need to introduce additional definitions of other binary preference operators; all of the remaining axioms are valid as well in Multi-Attribute Preference Logic when such definitions are added. Below we use that $\wedge$ and $\vee$ bind their arguments stronger than $\to$ to be able to remove some brackets.

Proposition 7.1. We have the following validities:

1. $\models E_i\varphi \leftrightarrow U_i\varphi$ for $\varphi \in \mathcal{L}_0$.
2. $\models \varphi <_{\forall\forall} \psi \wedge U(\xi \to \psi) \to \varphi <_{\forall\forall} \xi$
3. $\models \varphi <_{\forall\forall} \psi \wedge U(\xi \to \varphi) \to \xi <_{\forall\forall} \psi$
4. $\models \varphi <_{\forall\forall} \psi \wedge \psi <_{\forall\forall} \xi \wedge E\xi \to \varphi <_{\forall\forall} \xi$
5. $\models U\neg\varphi \vee U\neg\psi \to \varphi <_{\forall\forall} \psi$
6. $\models \varphi <_{\forall\forall} \psi \to U(\varphi <_{\forall\forall} \psi)$

What Multi-Attribute Preference Logic adds to binary preference logic are names for objects, and most importantly, the $\Box^{\neq}$ operator that allows us to define clusters (see Section 7.2.2) that represent desirable attributes. All of the modal operators $\Box^{=}, \Box^{<}, \Box^{\neq}$ and U are normal modal operators and satisfy the K axiom. In addition, we prove some properties of the $\Box^{=}$ and $\Box^{\neq}$ operators (some of the more obvious axioms have not been listed below). Proposition 7.2.3 shows that Multi-Attribute Preference Logic is related to the logic of only knowing, see [60].

Proposition 7.2. We have:

1. $\models \Box^{=}\Box^{\neq}\varphi \leftrightarrow \Box^{\neq}\varphi$
2. $\models \Box^{=}\Box^{<}\varphi \leftrightarrow \Box^{<}\varphi$
3. $\models \Box^{=}\varphi \to \neg\Box^{\neq}\varphi$ where $\neg\varphi \in \mathcal{L}_0$ is consistent

Proof. We prove item 3. Suppose $\Box^{=}\varphi$ is true at world w. Then φ is true in all worlds $v \sim w$. Since the truth of objective formulae is the same within an object, φ is also true in every world $u \in \{[v]_V \mid w \sim v\}$. Since $\neg\varphi$ is a consistent objective formula and all valuations are present in the model, $\neg\varphi$ must be true in some world in the model. So there must be some world in $\overline{\{[v]_V \mid w \sim v\}}$ that satisfies $\neg\varphi$, so we have $\neg\Box^{\neq}\varphi$ at world w. $\Box$

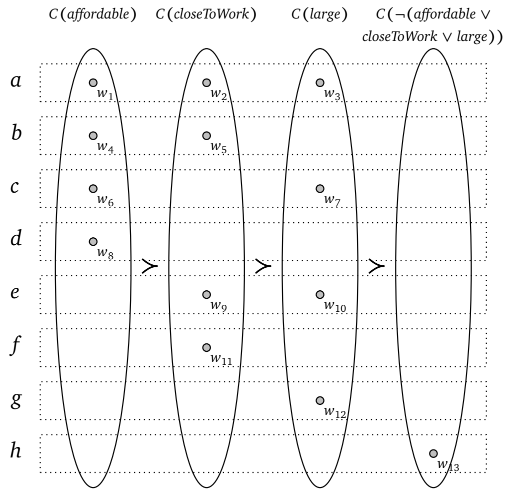


Figure 7.2: Visualization of an MPL model

7.2.2 Clusters

The total pre-order $\precsim$ in a multi-attribute preference model induces a strict linear order on sets of worlds, which we call clusters. Formally, a cluster is an equivalence class induced by $\precsim$. Intuitively, such clusters represent the properties or attributes considered relevant for deriving object preferences. The order on clusters induced by $\precsim$ represents a property ranking, i.e. the relative importance of one property compared to another. The relation between objects and properties may now be clarified as follows. The idea is that if an object has a particular property it should be represented within the cluster of worlds that represents the property. Technically, this is realized by making sure that (at least) one of the copies of a world that identifies the object is an element of the cluster that represents the property. The worlds that identify an object act as representatives for the object within a certain cluster and thus indicate that the object has that property. As clusters are disjoint and objects may have multiple properties, this also explains the need for introducing copies of worlds.

Definition 7.5. (Cluster) Let $\precsim$ be a total pre-order on W. A *cluster* c is an equivalence class induced by $\precsim$, i.e. $c = [w]_{\precsim} = \{v \mid w \sim v\}$ for some $w \in W$.

Example 7.2. The relation between clusters (properties) and sets of copies (objects) is visualized in Figure 7.2 (this is a model of the theory in Example 7.4). The ellipses (columns) represent the clusters or properties and the boxes (rows) represent objects. Objects in this case are supposed to be houses. For example, the house labelled b consists of two worlds, w_4 and w_5. As these worlds are part of the same object, they

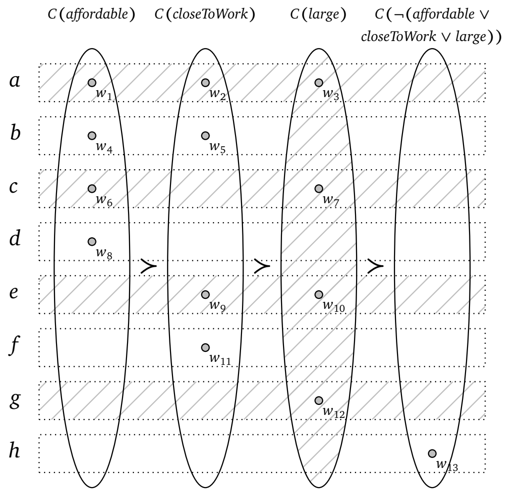


Figure 7.3: Visualization of an MPL model. All worlds where *large* is true are in the shaded section.

must be copies of each other. One of these worlds, w_4, is also part of the cluster representing the property of being affordable. This means that house b is affordable, as *affordable* is true at w_4 (and thus also at w_5). Similarly, it follows that house b is close to work, a property that is true at w_5 (and thus at w_4). As there is no world that is part of object b as well as in the cluster representing the property *large*, house b is not large. The ranking of the properties is indicated by the $\prec$ symbol: property *affordable* is more important than *close to work* which in turn is more important than *large*. As a result, in any natural preference ordering based on this ranking one would expect house b to be preferred over house c.

The modality $\Box^=$ can be used to express a property of a cluster. For example, $E\,\Box^=\,\varphi$ expresses that there is a cluster where φ is true everywhere. $\Box^=\varphi$ expresses that at least φ is true in the cluster. In Figure 7.2, for example, in the third cluster we have that $\Box^=$*large* is true. This means that every object that is represented by a world in this cluster is *large*. But we also want every object that is *large* to be represented in the cluster. To specify this, we use the modality $\Box^{\neq}$. We can now explain why simply defining the truth of $\Box^{\neq}\varphi$ in terms of truth of φ in all worlds that are not equally ranked to the current one does not work. The point is that there may be copies v of worlds w that have a different ranking than world w. As copies have the same truth assignment, at such copies a propositional formula φ would be assigned the same truth value. This is illustrated in Figure 7.3, where *large* is true in all worlds in the shaded area. The key observation here is that worlds of a particular ranking identify a set of objects, i.e. copies of these worlds which must be part of these objects (by Definition 7.2 of an object). This is why $\Box^{\neq}\varphi$ evaluates φ at all objects, or, more precisely, the worlds that define these objects, that are not identified by any of the worlds that have the same ranking as the current one.

By combining both operators we are able to characterize a cluster. For the third cluster in Figure 7.2, we have that $\Box^{=}large \wedge \Box^{\neq}\neg large$ where *large* exactly characterizes the cluster. The characterization of a cluster by φ is abbreviated as $C\varphi$, and defined by:

$$C\varphi \;::=\; \Box^{=}\varphi \wedge \Box^{\neq}\neg\varphi$$

φ is true for all objects identified by (worlds in) the cluster and not true in all worlds that identify other objects. As an object may consist of several copies to represent that it has various properties represented by different clusters, copies of such worlds outside the cluster need to be excluded in the evaluation of $\neg\varphi$ which explains the truth condition for $\Box^{\neq}$.

Proposition 7.3 shows that properties and objects are related in such a way that object preferences can be derived. The first item of the proposition states that if there is an object that has property φ and the current world identifies a cluster characterized by φ, then within the cluster there is a world that is named i, i.e. identifies the object i. The second item states that the converse is true for an object that does not satisfy a property φ that characterizes a cluster. That is, if object i does not satisfy φ and the current world identifies a cluster characterized by φ, then no world that identifies the object labelled i is part of that cluster. The third item generalizes the first item. It states that if there is a cluster characterized by φ, and there is an object named i that satisfies φ, then there is an i-world in that cluster. The last item states that when a world satisfies $C(\varphi)$, then all worlds within the same cluster satisfy $C(\varphi)$.

Proposition 7.3. We have:

1. $\models C(\varphi) \wedge E_i\varphi \rightarrow \Diamond^{=} i$
2. $\models C(\varphi) \wedge \neg E_i\varphi \rightarrow \neg \Diamond^{=} i$
3. $\models EC(\varphi) \wedge E_i\varphi \rightarrow E_iC(\varphi)$
4. $\models C(\varphi) \rightarrow \Box^{=}C(\varphi)$

Proof. We prove item 1. Suppose $\mathcal{M}, w \models C(\varphi)\wedge E_i\varphi$. This means that $\mathcal{M}, w \models \Box^{\neq}\neg\varphi$. By the truth definition for $\Box^{\neq}$, this is equivalent to $\forall u \in \overline{\bigcup\{[v]_V \mid w \sim v\}} : \mathcal{M}, u \models \neg\varphi$. By the definition of $E_i\varphi$ we must also have a world u' such that $\mathcal{M}, u' \models i \wedge \varphi$. This means that we cannot have $u' \in \overline{\bigcup\{[v]_V \mid w \sim v\}}$ and we have that $u' \in \bigcup\{[v]_V \mid w \sim v\}$. It follows that $u' \in [v]_V$ for some $v \sim w$; as u' must be a copy of v this means that we have $\mathcal{M}, v \models i$ and, by the truth definition for $\Diamond^{=}$, we have $\mathcal{M}, w \models \Diamond^{=} i$. □

The operator C provides exactly what we need to define property rankings. Semantically, we have already seen that the pre-order $\precsim$ induces a strict linear order on clusters. The formula $C\varphi$ allows us to express that a cluster is characterized by a formula φ. Using this operator and the binary preference operator $<_{\forall\forall}$ we can express that property ψ (represented by a cluster) is ranked higher than another property φ (represented by another cluster) by $C\varphi <_{\forall\forall} C\psi$. For example, in Figure

7.2, we have $C(large) <_{\forall\forall} C(closeToWork) <_{\forall\forall} C(affordable)$. By combining this with specifications of particular preferences orderings and statements that an object has a particular property (cf. Proposition 7.3), this will allow the derivation of object preferences from a property ranking.

7.3 Preference orderings

In this Section, we show how to use Multi-Attribute Preference Logic to define multi-attribute preference orderings derived from property rankings. Coste-Marquis *et al.* [44] describe three frequent orderings based on prioritized goals: best-out, discrimin and leximin ordering. Brewka [33] defines a preference language in which different basic preference orderings can be combined and identifies four 'fundamental strategies' for deriving preferences from what he calls a ranked knowledge base: $\top$, κ, $\subseteq$ and $\#$. As best-out is the same as κ, discrimin is $\subseteq$, and leximin is $\#$, we will base the remainder of our discussion on Brewka [33].

We first informally introduce these orderings and then present definitions for each of them in the logic. Section 7.4 presents the definitions of [33] and a proof that the definitions in Multi-Attribute Preference Logic match those provided in [33]. The advantage of defining preference orderings in a logic instead of providing set-theoretical definitions is that it formalizes the reasoning about object preferences. From a practical point of view, the logic allows us to provide rigorous formal proofs for object preferences derived from property rankings. From a theoretical point of view, it provides the tools to reason *about* preference orderings and allows, for example, to prove that whenever an object is preferred over another by the $\top$ strategy it also is preferred by the # strategy (see Proposition 7.4 below).

The two orderings $\subseteq$ and # first consider the most important property. If some object has that property and another does not, then the first is preferred over the second. So in the example, both $house_1$ and $house_2$ would be preferred over $house_3$. If two houses both have the property or if neither of them has it, the next property is considered. $house_1$ and $house_2$ are both affordable, but $house_1$ is close to work and $house_2$ is not, so $house_1$ would be preferred over $house_2$. Note that although $house_3$ satisfies two properties and house $house_2$ only satisfies one property, $house_2$ is still preferred over $house_3$ because the single property of $house_2$ is considered more important than both properties of $house_3$. The $\subseteq$ and # orderings only differ if multiple properties are equally important. As we will make the assumption that no two properties can have the same importance, we will not discuss the difference and only refer to the # ordering in the following.

The $\top$ ordering looks at the highest ranked or most important property that *is* satisfied. If that property of one object is ranked higher than that of another object, then the first object is preferred over the second. If those properties are equally ranked, then both objects are equally preferred. In our running example, $house_1$ and $house_2$ are both preferred over $house_3$, since the property ranked highest that is satisfied by both $house_1$ and $house_2$ is *affordable*, and this property is ranked higher than the highest ranked property satisfied by $house_3$, i.e. *closeToWork*. Since the most important property satisfied by $house_1$ is the same as the most important property satisfied by $house_2$, $house_1$ and $house_2$ are equally preferred.

The κ ordering looks at the most important property that *is not* satisfied. If that property of one object is less important than the property of another object, then the first object is preferred over the second. If those properties are equally important, then both objects are equally preferred. In our running example, the highest ranked property that is not satisfied by $house_1$ is *large*, that of $house_2$ is *closeToWork* and that of $house_3$ is *affordable*. Since *large* is the least important property of these properties, $house_1$ is preferred over both other houses. As *closeToWork* is less important than *affordable*, $house_2$ is preferred over $house_3$.

All preference orderings introduced can be defined in multi-attribute preference logic. We use $pref^s_{\sim}(i,j)$ to stand for: object i is weakly preferred over object j according to strategy s, where s is one of $\top$, κ and $\#$; $pref^s(i,j)$ is used to express strict preference.

Definition 7.6. (Preference Orderings) $pref^{\kappa}(i,j)$, $pref^{\kappa}_{\sim}(i,j)$, $pref^{\#}(i,j)$, $pref^{\#}_{\sim}(i,j)$, $pref^{\top}(i,j)$ and $pref^{\top}_{\sim}(i,j)$ are defined by:

$$\begin{array}{lcl}
pref^{\top}(i,j) & ::= & E(i \wedge \neg \Diamond^{=} j \wedge \Box^{<}(\neg i \wedge \neg j)) \\
pref^{\top}_{\sim}(i,j) & ::= & pref^{\top}(i,j) \vee U((\Diamond^{=} i \wedge \Box^{<}\neg i) \leftrightarrow (\Diamond^{=} j \wedge \Box^{<}\neg j)) \\
pref^{\kappa}(i,j) & ::= & E(i \wedge \neg \Diamond^{=} j \wedge \Box^{<}(\Diamond^{=} i \wedge \Diamond^{=} j)) \\
pref^{\kappa}_{\sim}(i,j) & ::= & pref^{\kappa}(i,j) \vee U((\neg \Diamond^{=} i \wedge \Box^{<} \Diamond^{=} i) \leftrightarrow (\neg \Diamond^{=} j \wedge \Box^{<} \Diamond^{=} j)) \\
pref^{\#}(i,j) & ::= & E(i \wedge \neg \Diamond^{=} j \wedge \Box^{<}(\Diamond^{=} i \leftrightarrow \Diamond^{=} j)) \\
pref^{\#}_{\sim}(i,j) & ::= & pref^{\#}(i,j) \vee U(\Diamond^{=} i \leftrightarrow \Diamond^{=} j)
\end{array}$$

To understand these definitions, recall that we say that a world identifies an object when it is part of that object and the object consists of copies of one and the same world. These copies are used to represent that an object has a property present in a property ranking. In Figure 7.2, for example, world w_7 is a representative of object c for the property *large*. Thus, the formula $E_i \neg \Diamond^{=} j$ may be read as 'object i has a property that object j does not have'. Similarly, $\Diamond^{<} i$ can be read as 'there is a more important property (than the current one) that object i has'. These readings may help explain the definitions. $pref^{\top}(i,j)$ may be read as 'there is a property such that i has it and j does not, and for all more important properties, neither i nor j has any of them'. The second disjunct in the definition of $pref^{\top}_{\sim}(i,j)$ defines when two objects are equally preferred with respect to $\top$, and may be read as 'if there is a property that i has, but i does not have any more important properties, then j has that property too and does not have any more important properties either, and vice versa'. Similar readings can be provided for the other preference operators.

Proposition 7.4 shows that the relation between weak and strict preference is as usual, and, moreover, a strict preference according to $\top$ or κ implies a strict preference according to $\#$.

Proposition 7.4. We have:

1. $\models pref^s(i,j) \leftrightarrow pref^s_\sim(i,j) \wedge \neg pref^s_\sim(j,i)$ for $s \in \{\top, \kappa, \#\}$
2. $\models pref^\top(i,j) \rightarrow pref^\#(i,j)$
3. $\models pref^\kappa(i,j) \rightarrow pref^\#(i,j)$

Example 7.3. Given the model of Figure 7.2, we can derive that $pref^\#(b,d)$. By definition, this is the case when $E(b \wedge \neg \Diamond^= d \wedge \Box^<(\Diamond^= b \leftrightarrow \Diamond^= d))$ is true. This means that there must be a world w that is named b that has no equally ranked world named d, and, moreover, for every higher ranked world v there is an equally ranked world named b if and only if there is an equally ranked world with name d. By inspection of Figure 7.2, world w_5 fits the description.

7.4 MPL defines ranked knowledge bases

Here we prove that the preference orderings of Definition 7.6 define those of Brewka [33]. Brewka calls property rankings *ranked knowledge bases*, defined as follows:

Definition 7.7. (Ranked knowledge base) A *ranked knowledge base* (RKB) is a set $F \subseteq \mathcal{L}_0$ of objective formulae together with a total pre-order $\geq$ on F. Ranked knowledge bases are represented as a set of ranked formulae (f,k), where f is an objective formula and k, the rank of f, is a non-negative integer such that $f_1 \geq f_2$ iff $rank(f_1) \geq rank(f_2)$. That is, higher rank is expressed by higher indices.

In the setting of [33], comparing objects given a ranked knowledge base means comparing *truth assignments* which represent these objects, analogously to the representation of the three houses used in Figure 7.1. It is easy to see that this example is represented by the following ranked knowledge base: $\{(\mathit{affordable}, 3), (\mathit{closeToWork}, 2), (\mathit{large}, 1)\}$.

Object preferences can be derived in multiple ways from a ranked knowledge base. In order to define these strategies, some auxiliary definitions are introduced next. Below, $K^n(m)$ denotes the set of properties of a certain rank n that are satisfied with respect to truth assignment m; $maxsat^K(m)$ denotes the highest rank associated with the properties that are satisfied by assignment m, and $maxunsat^K(m)$ denotes the highest rank associated with the properties that are not satisfied by m.

Definition 7.8. Let K be a ranked knowledge base and $m \in 2^{At}$.

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

Using these auxiliary definitions, preference orderings $m_1 \geq^K_s m_2$ are defined which mean that object (truth assignment) m_1 is (weakly) preferred over object m_2 according to strategy s.

Definition 7.9. (Preference orderings) Let K be a ranked knowledge base. Then the following preference orderings over truth assignments are defined:

- $m_1 \geq^K_\top m_2$ iff $maxsat^K(m_1) \geq maxsat^K(m_2)$.
- $m_1 \geq^K_\kappa m_2$ iff $maxunsat^K(m_1) \leq maxunsat^K(m_2)$.
- $m_1 \geq^K_\# m_2$ iff $|K^n(m_1)| = |K^n(m_2)|$ for all n, or there is n s.t. $|K^n(m_1)| > |K^n(m_2)|$, and for all $j > n : |K^j(m_1)| = |K^j(m_2)|$.

To simplify, we make the assumption here that different properties cannot have the same ranking. In that case, the set of all satisfied properties of a given rank is a singleton set or the empty set, we have that $\geq$ is a strict linear order on F - also denoted by $>$, and, as a result, the $\subseteq$ and $\#$ orderings coincide. We also assume that properties in a ranked knowledge base are consistent. Finally, we may assume that a ranked knowledge base does not contain logically equivalent properties with different ranks since such occurrences except for the one ranked highest can be discarded as it has no influence on any of the preference orderings.

Definition 7.10. (Translation function) The function τ translates ranked knowledge bases $K = \langle F, \geq \rangle$ and truth assignments m to formulae and is defined by:

- $\tau(K) ::= \bigwedge\{EC(\varphi) \mid \varphi \in F\} \wedge$
  $U(\bigvee\{C(\varphi) \mid \varphi \in F \text{ or } \varphi = \neg \bigvee\{\chi \mid \chi \in F\}\}) \wedge$
  $\bigwedge\{C(\varphi) <_{\forall\forall} C(\psi) \mid \varphi, \psi \in F \;\&\; \psi > \varphi\} \wedge$
  $\bigwedge\{C(\neg \bigvee\{\varphi \mid \varphi \in F\}) <_{\forall\forall} \psi \mid \psi \in F\}$
- $\tau_{name}(m) \in Nom$
- $\tau(m) ::= \bigwedge\{E_i\varphi \mid m \vDash \varphi\} \cup \{\neg E_i \varphi \mid m \nvDash \varphi\}$ with $i = \tau_{name}(m)$

The translation of a ranked knowledge base K expresses that for each property φ in K, there exists a corresponding cluster by $C\varphi$, that there are no other clusters than those specified by the properties, and one extra cluster for the case in which none of the properties is satisfied. It forces the ranking of these clusters to be the same as the property ranking induced by K, with the added extra cluster as least important one. The translation also associates an object name with a truth assignment and states for each property whether the object (truth assignment) has the property or not.

Example 7.4. Using the translation function, and assuming that $\tau_{name}(house_1) = b$, $\tau_{name}(house_2) = d$ and $\tau_{name}(house_3) = e$, the RKB $\{(affordable, 3), (closeToWork, 2), (large, 1)\}$ translates into:

1. $E(C(affordable)) \wedge E(C(closeToWork)) \wedge E(C(large))$
2. $U(C(affordable) \vee C(closeToWork) \vee C(large) \vee C(\neg(affordable \vee closeToWork \vee large)))$
3. $C(\neg(affordable \vee closeToWork \vee large)) <_{\forall\forall} C(large) <_{\forall\forall} C(closeToWork) <_{\forall\forall} C(affordable)$
4. $E_b(affordable) \wedge E_b(closeToWork) \wedge \neg E_b(large)$
5. $E_d(affordable) \wedge \neg E_d(closeToWork) \wedge \neg E_d(large)$
6. $\neg E_e(affordable) \wedge E_e(closeToWork) \wedge E_e(large)$

A model of this theory is shown in Figure 7.2. Although only objects b, d and e are specified in the theory, for illustrative reasons this model contains all possible objects (there is a world, and hence an object, for every possible valuation of the three propositional atoms). Every property has its own cluster, which means that every object satisfying that property has a world in that cluster, and that every world in that cluster satisfies that property. No worlds exist outside the four specified clusters, and the order among clusters is fixed. The only ways a model of this theory can be structurally different from the one shown are by removing objects that are not b, d or e (but then all worlds belonging to that object have to be removed at once), or by adding more worlds, but only at the same 'places' as the worlds shown.

Theorem 7.1 shows that every multi-attribute preference model that is a model of the translation of a particular RKB yields the same preference ordering as the original RKB.

Theorem 7.1. $m_1 \geq_s^K m_2$ iff $\models \tau(K) \wedge \tau(m_1) \wedge \tau(m_2) \rightarrow \mathit{pref}_{\sim}^{s}(\tau_{name}(m_1), \tau_{name}(m_2))$ where $s \in \{\top, \kappa, \#\}$.

Proof. Assume that $\tau_{name}(m_1) = i$ and $\tau_{name}(m_2) = j$, and observe that the translation of $K = \langle F, \geq \rangle$ is equivalent to:
(1) $C(\neg(f_1 \vee \ldots \vee f_n)) <_{\forall\forall} C(f_1) <_{\forall\forall} \ldots <_{\forall\forall} C(f_n)$,
(2) $\forall f \in F : E(C(f))$ and
(3) $U(C(f_1) \vee \ldots \vee C(f_n) \vee C(\neg(f_1 \vee \ldots \vee f_n)))$.
For brevity, we only prove the left to right direction for the case $m_1 >_\kappa^K m_2$. Then we have $maxunsat^K(m_1) < maxunsat^K(m_2)$ and $maxunsat^K(m_2) > -\infty$, so there is a formula f_k in F such that
(4) $m_2 \not\models f_k$,
(5) $m_1 \models f_k$ and
(6) $\forall f' > f_k : m_1 \models f'$ & $m_2 \models f'$.
Applying the translation function τ, we then get:
(4) $\neg E_j f_k$,
(5) $E_i f_k$ and
(6) $\forall f' > f_k : E_i f' \wedge E_j f'$.
From (5), (2) and Prop. 7.3.3 it then follows that
(8) $E_i C(f_k)$.
From (8), (4) and Prop. 7.3.2 it follows that
(9) $E_i \neg \Diamond^= j \wedge C(f_k)$.
And from (6) and Prop. 7.3.1 it follows that
(10) $\forall f' > f_k : \Diamond^= i \wedge \Diamond^= j$.
Using (1) and (3) we obtain
(11) $C(f_k) \rightarrow \Box^<(C(f_{k+1}) \vee \ldots \vee C(f_n))$.
From (10) and (11) we obtain
(12) $C(f_k) \rightarrow \Box^< \Diamond^= i \wedge \Box^< \Diamond^= j$.
Then (9) and (12) can be combined into $E(i \wedge \neg \Diamond^= j \wedge \Box^<(\Diamond^= i \wedge \Diamond^= j))$, which is the definition of $\mathit{pref}^\kappa(i, j)$. $\Box$

Example 7.5. We now show how to formally derive a preference statement from the formulae obtained by translating a ranked knowledge base in Example 7.4. As an illustration, we show that $pref^{\kappa}(b, d)$ can be derived.

From (7.4.4) $E_b(closeToWork)$, (7.4.1) $E(C(closeToWork))$ and Proposition 7.3.3 we obtain

(1) $E_b C(closeToWork)$.

From (7.4.5) $\neg E_d(closeToWork)$ and Proposition 7.3.2 it follows that

(2a) $C(closeToWork) \rightarrow \neg \Diamond^{=} d$.

From 7.4.3 and 7.4.2 we can derive that

(2b) $C(closeToWork) \rightarrow \Box^{<} C(affordable)$.

By combining (1), (2a) and (2b) we derive

(3) $E_b(\neg \Diamond^{=} d \wedge \Box^{<} C(affordable))$.

Now, from Proposition 7.3.1, (7.4.4) $E_b(affordable)$ and (7.4.5) $E_d(affordable)$, we derive

(4a) $C(affordable) \rightarrow \Diamond^{=} b$ and

(4b) $C(affordable) \rightarrow \Diamond^{=} b$.

Using (3), (4a), and (4b), we obtain $E_b(\neg \Diamond^{=} d \wedge \Box^{<}(\Diamond^{=} b \wedge \Diamond^{=} d))$, which is the definition of $pref^{\kappa}(b, d)$.

7.5 Conclusion

In this paper we introduced a modal logic for qualitative multi-attribute preferences. The logic is based on Girard's binary preference logic [58], but extends this logic with objects and clusters that introduce the possibility to reason explicitly about multiple attributes. We showed that Multi-Attribute Preference Logic is expressive enough to define various natural preference orderings based on property rankings [33, 44]. The additional value of the logic is that it is possible to reason about these different preference orderings within the logic. This means we cannot only reason about which objects are preferred according to a certain ordering, but also about the relation between different orderings as is shown in Proposition 7.4.

One possible extension to Multi-Attribute Preference Logic is the introduction of indices for different agents. In this way, distinct preference orderings for several agents can be expressed. This introduces the possibility to reason about properties such as pareto-optimality of objects (an object is pareto-optimal if there is no other object that is better for at least one agent and not worse for the other agents), which is useful in the context of e.g. joint decision making or negotiation.

We have made the assumptions that attributes are binary, and that priority orderings are total linear orders. In future work we plan to investigate how we can loosen these assumptions. For example, if multiple attributes can have the same importance, the # and $\subseteq$ orderings will differ and we will be able to encode trade-offs between attributes.

Our main concern in this paper has been the expressiveness of Multi-Attribute Preference Logic. Other questions such as a complete axiomatization of the logic, succinctness and complexity remain future work. We plan to develop a reasoning system in which agents can reason about qualitative multi-attribute preferences in various settings. In our future work we will focus more on the reasoning mechanism and how different domains can be modelled accurately in our approach.

A more detailed comparison of Multi-Attribute Preference Logic with other preference logics such as Qualitative Choice Logic [34] is planned. Other areas for future work concern the representation of dependent properties and the relation of Multi-Attribute Preference Logic to e.g. CP-nets [29].

Chapter 8

Conclusion

8.1 Results

In this thesis we have developed a framework for the representation of, and reasoning about qualitative multi-criteria preferences. In the Introduction we specified the particular research questions that this thesis addresses. In this section we discuss the results for every question in turn.

1. *How can argumentation be used to reason about qualitative multi-criteria preferences?*

Several argumentation frameworks, each including a logical language, a set of inference rules, and a defeat relation, have been defined in this thesis. In Section 2.3 of Chapter 2 we presented a basic argumentation framework to reason about qualitative multi-criteria preferences. This framework uses a simple definition of objects and preferences between them. Objects are defined as value assignments to a set of attributes which are all binary. For preference, a version of the lexicographic ordering is used where the criteria are the same as the attributes that define the objects, and the importance between them is a total preorder. This basic framework provides the 'proof of concept' that argumentation is a suitable tool to reason about qualitative multi-criteria preferences. All other argumentation frameworks in this thesis build on this first framework.

In Section 2.5 in Chapter 2 we presented an argumentation framework that implements the proposed strategy to derive preferences in case of incomplete information about the objects to be compared. In Section 2.6 in Chapter 2 we proposed an epistemic argumentation framework to reason about background knowledge with different degrees of certainty. Such information is used in the two argumentation frameworks presented in Section 2.7 in Chapter 2 that implement the proposed strategies to derive preferences in case of uncertain information about the objects to be compared.

In Section 3.6 in Chapter 3 we presented an argumentation framework for reasoning about qualitative multi-criteria preferences that uses a definition of preference that is more abstract compared to the one used in the basic framework in Chapter 2. Here, criteria can also be derived attributes (in particular underlying interests), and the importance between them can be any preorder, thus generalizing both the lexicographic variant used in Chapter 2 and ceteris paribus preference. This argumentation framework also offers a basic means of reasoning about facts by providing an inference scheme for defeasible modus ponens.

Finally, in Chapter 5 we defined an argumentation framework to reason about Qualitative Preference Systems, the general framework for the representation of qualitative, multi-criteria preferences that was introduced in Chapter 4. Here, the variables that define outcomes are no longer assumed to be binary but can have arbitrary domains. The definition of preference is also more general than before, with three types of criteria that can be combined in a layered structure. Moreover, this chapter proposed an extension of the argumentation framework in which it is possible to reason with background knowledge to derive information about the values of variables by default. This is a useful feature for which argumentation is especially suitable.

2a. *How can qualitative multi-criteria preferences be derived when information about the outcomes is incomplete?*

In Section 2.4 in Chapter 2 we discussed some naive strategies of dealing with preferences between objects for which it is not known for every attribute whether it is true or false. From the limitations of these strategies, we identified two desired properties for strategies handling preferences based on incomplete information: decisiveness and safety. We then proposed an adequate strategy that is both decisive and safe, based on the notion of least and most preferred completions of objects. This definition generalizes the simple preference definition used in the first part of the chapter: if all information is complete, it results in the same preferences.

2b. *How can qualitative multi-criteria preferences be derived when information about the outcomes is uncertain?*

In Section 2.6 in Chapter 2 we explored how uncertain (defeasible) information can be represented ordinally using certainty levels (degrees of belief). In Section 2.7 we discussed some purely qualitative strategies to reason about preferences between objects for which the truth or falsehood of attributes is uncertain. We then defined a compensatory strategy and a safer compensatory strategy, which are based on the notion of subjective probability. The safer compensatory strategy generalizes both the compensatory strategy and the decisive and safe strategy for handling incomplete information from Section 2.4.

3. *What kind of attributes should be chosen as criteria?*

In Chapter 3 we argued that instead of issues (the attributes that define negotiation outcomes), the negotiators' underlying interests should be chosen as criteria, especially if the issues are not preferentially independent. We showed that using interests as criteria is more flexible than modelling conditional preferences, and provides a better explanation of the derived preferences. In Chapter 3 all attributes are binary, but in Section 4.3.1 in Chapter 4 we illustrated some possibilities to model interests when attributes have arbitrary domains.

4a. *How can a general framework for the representation of qualitative multi-criteria preferences over multi-attribute domains be defined?*

In Section 4.2 in Chapter 4 we presented a general framework for the representation of qualitative, multi-criteria preferences, called Qualitative Preference Systems (QPS). The model is more general than the ones presented in the previous chapters, in which attributes and criteria were assumed to be binary. Here, outcomes are defined as value assignments to a set of variables which can have arbitrary domains. The framework includes a knowledge base that serves two purposes: to impose (hard) constraints and to define new (abstract) concepts. Three types of criteria are defined. Simple criteria derive a preference relation over outcomes from a preference relation on the values of a single variable. Multiple criteria can be combined in a cardinality criterion, which is based on counting the number of criteria that support a preference, or in a lexicographic criterion, which is based on priority. Together, all used criteria form a layered structure called a criterion tree.

4b. *How expressive is the proposed framework?*

In Section 4.3 in Chapter 4 we showed that QPS can model conditional preferences and underlying interests, goal-based preferences, and bipolar preferences. We also compared the QPS framework in detail with two other, well-known approaches that are representative for a large number of purely qualitative approaches to modelling preferences, namely Logical Preference Description language [33] and CP-nets [29]. We showed that the Logical Preference Descriptions can be translated to the QPS framework (with the exception of the disjunction operator which is not natural as it does not satisfy independence of irrelevant alternatives and unanimity with abstentions [8]), and provided an order preserving translation of acyclic CP-nets into QPSs which satisfies the ceteris paribus condition. In addition, we showed that these translations are size preserving, i.e. the resulting QPSs provide a representation that is as succinct as the LPD or CP-net representation.

4c. *How expressive are binary goals as criteria?*

In Section 4.4 in Chapter 4 we showed that any QPS (including simple criteria ranging over multi-valued variables) can be translated to an equivalent and just as succinct goal-based QPS where all simple criteria have been replaced by goals. Moreover, in Section 4.5 we showed that goal-based QPSs allow more fine-grained updates of the criterion tree because goals relating to different variables can be interleaved.

5a. *How can a preference model be explained?*

In Section 6.3 in Chapter 6 we proposed to use the structure of a QPS criterion tree to generate explanations for the resulting preferences between outcomes. We used the intuition that preferences can be explained by the criteria that are deciding in the overall preference. Explanations were proposed for every kind of preference by every type of criterion.

5b. *How can explanations of preference provided by a user be used by a system to update the preference model?*

In Section 6.4 in Chapter 6 we provided detailed interaction diagrams that specify how the system should react to an explanation, given by the user, of a preference that does not follow from the current model. There are basically two possibilities: to ask the user a follow-up question or to update the preference model. In the latter case, the updated preference model will not only support the same preference as stated by the user, but also generate the same explanation for it.

6. *How can modal logic be used to reason about qualitative preferences and the relations between preference orderings?*

In Chapter 7 we introduced a modal logic, called Multi-Attribute Preference Logic (MPL), that provides a language for expressing several strategies to qualitatively derive a preference between objects (outcomes) from property (attribute) rankings. Objects here are defined as specific sets of possible worlds (propositional models) that share the same truth assignments. Preferences are derived from a set of desired properties (propositional formulas) that are ranked according to importance. Three different strategies from the literature on prioritized goals [44, 33] to derive preferences from property rankings are modelled. The additional value of the logic is that it is possible to reason about these different preference orderings within the logic. This means we cannot only reason about which objects are preferred according to a certain ordering, but also about the relation between different orderings.

8.2 Directions for future research

First of all, we would like to implement the Qualitative Preference Systems framework, together with the proposed argumentation framework and the explanation facility. An important issue that we would have to deal with is computational complexity, and the design of efficient algorithms. When an implementation of the system is available, this would provide the opportunity to perform user experiments to test how well the framework can represent human preferences and how natural the generated explanations are. To this end, preference elicitation methods specifically tailored to the QPS framework have to be developed. One approach to update preference models was proposed in Chapter 6, but it needs to be developed further. Specifically, some issues of consistency have to be investigated. For example, priorities have to remain partial orders, value preferences have to remain preorders, and previous preferences and explanations expressed by the user should remain valid. We also believe that the argumentation-based approach 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 he sees fit. To facilitate this elicitation process, more research is needed on how our framework can support a user, e.g. by indicating which information is still missing.

Next, the nature of argumentation as a form of defeasible reasoning can be exploited more. In Chapter 4, the elements of a QPS were always completely specified. In Chapter 5, we showed how factual information about outcomes can be derived by default in an argumentation framework, thus making it more flexible in case of incomplete outcome specifications. The framework can be made even more flexible if also preferential information about criteria, value preferences and priorities can be derived from other information. This would be especially useful when modelling preferences of others (e.g. the opponent in negotiation), where it is not realistic that all relevant information is available, but some of it may be inferred by default.

Also, the explantation facility introduced in Chapter 6 can be developed further. In some instances it may be necessary to explain facts about the outcomes involved in a preferential comparison, e.g. to explain why they do or do not satisfy a particular goal. Explanation of knowledge and reasoning is a separate field of study that may provide solutions to this issue. Also, the explanation of preferences may be part of a larger picture, for example in recommendation, decision making or planning. It would be interesting to investigate how the proposed preference explanation mechanism can be embedded in other explanation mechanisms that explain a certain decision or recommendation in terms of the underlying preferences.

The ultimate goal of the Pocket Negotiator project is to develop a negotiation support system. To incorporate qualitative preferences in this system, qualitative negotiation strategies have to be developed, as opposed to the utility-based approaches currently in use. One option is to use qualitative preferences as input, but transform them into a quantitative representation that is used in the negotiation strategy, as is done for example in [10] for incomplete preferences expressed by a CP-net. Another option is to develop new negotiation strategies that deal with qualitative, and possibly incomplete, preferences directly. Different protocols and strategies can be defined, for example only exchanging offers or also communicating through arguments. Simulation experiments have to show which strategies perform best in different settings, which could vary with e.g. the kind of preferences of the agents, the distribution of knowledge among the agents, or the trustworthiness of agents. In particular, it is interesting to look at how the exchange of information about (qualitative) preferences, background knowledge and underlying interests affects the performance of an agent negotiating on behalf of a human user.

Summary

Qualitative multi-criteria preference representation and reasoning

The research reported on in this thesis is part of a larger research project that aims to develop a negotiation support system called the Pocket Negotiator. This thesis focuses on the question how such a system can represent and reason about a user's preferences between the possible outcomes of a negotiation. In real-world negotiations, there are many negotiation issues which can have many different values, resulting in a large space of complex outcomes. A negotiation support system needs to have a model of the user's preferences over this outcome space. Although most current negotiation support systems use numerical measures such as utility to represent preferences, such quantitative preferences are hard to specify for human users, and so it would be more natural to model the user's preferences in a qualitative way. Moreover, due to the exponential size of the outcome space, it is not feasible to specify a preference ordering directly. Therefore, we aim to represent the preferences in a more compact way by aggregating multiple evaluation criteria that influence preference.

The main research objective of this thesis is to develop a framework for the representation of, and reasoning about such qualitative multi-criteria preferences. The thesis makes the following contributions.

- We propose strategies to derive preferences from incomplete or uncertain information about the objects to be compared. The decisive and safe strategy for incomplete information is based on the notion of least and most preferred completions of objects. The strategies for uncertain information are based on an ordinal representation of the certainty levels of facts.

- We argue that instead of negotiation issues, the negotiators' underlying interests should be chosen as criteria, especially if the issues are not preferentially independent. We show that the use of interests as criteria is more flexible than modelling conditional preferences, and provides a better explanation of the derived preferences.

- We present a general framework for the representation of qualitative, multi-criteria preferences, called Qualitative Preference Systems (QPS). The frame-

work defines outcomes as value assignments to a set of variables which can have arbitrary domains, includes a knowledge base that can impose (hard) constraints and define new (abstract) concepts, and defines three types of criteria that can be combined in a tree structure. We show that the QPS framework is expressive, as it can model conditional preferences and underlying interests, goal-based preferences, bipolar preferences, and preferences represented in two other well-known approaches that are representative for a large number of purely qualitative preference modelling approaches. Moreover, we show that the goal-based variant of QPS is just as expressive.

- For all proposed preference representation frameworks we define corresponding argumentation frameworks that include a logical language, a set of inference rules, and a defeat relation. Some of the argumentation frameworks also provide the possibility to reason with background knowledge to derive information about the values of variables by default.

- We propose a mechanism to generate explanations for preferences represented in a QPS. We use the intuition that preferences can be explained by the criteria that are deciding in the overall preference. Moreover, we show how a system can use user-provided explanations to update its current preference model.

- Finally, we introduce a modal logic, called Multi-Attribute Preference Logic (MPL), that provides a language for expressing several strategies to qualitatively derive a preference between objects from property rankings. Three such strategies from the literature on prioritized goals are modelled. The additional value of the logic is that it is possible to reason not only about which objects are preferred according to a certain ordering, but also about the relation between different orderings.

Samenvatting

Kwalitatieve multi-criteria voorkeuren - representatie en redeneren

Het onderzoek waarvan in dit proefschrift verslag wordt gedaan is onderdeel van een groter onderzoeksproject met als doel de ontwikkeling van een onderhandelondersteuningssysteem, genaamd de Pocket Negotiator. Dit proefschrift concentreert zich op de vraag hoe zo'n systeem de voorkeuren van een gebruiker tussen de mogelijke uitkomsten van een onderhandeling kan representeren en hoe het daarover kan redeneren. In realistische onderhandelingen zijn er veel onderhandelingskwesties, die veel verschillende invullingen kunnen krijgen, wat resulteert in een grote ruimte met complexe uitkomsten. Een onderhandelondersteuningssysteem moet een model hebben van de voorkeuren van de gebruiker wat betreft deze uitkomstruimte. Hoewel de meeste huidige onderhandelondersteuningssystemen gebruik maken van numerieke maten zoals utiliteit, zijn zulke kwantitatieve voorkeuren moeilijk te formuleren voor menselijke gebruikers. Het is natuurlijker om de voorkeuren van de gebruiker op een meer kwalitatieve manier te modelleren. Verder is het, vanwege de exponentiële grootte van de uitkomstruimte, niet haalbaar om een voorkeursordening direct te specificeren. Daarom willen we de voorkeuren op een compactere wijze representeren door meerdere evaluatiecriteria die de voorkeur beïnvloeden te combineren.

Het hoofdonderzoeksdoel van dit proefschrift is de ontwikkeling van een framework voor de representatie van, en het redeneren over zulke kwalitatieve voorkeuren die gebaseerd zijn op meerdere criteria. Het proefschrift levert de volgende bijdragen.

- We stellen strategieën voor om voorkeuren af te leiden uit onvolledige of onzekere informatie over de te vergelijken objecten. De besluitvaardige en veilige strategie voor onvolledige informatie is gebaseerd op het concept van minst en meest geprefereerde vervolledigingen van objecten. De strategieën voor onzekere informatie zijn gebaseerd op een ordinale representatie van de zekerheidsniveaus van feiten.

- We betogen dat in plaats van de onderhandelingskwesties, de onderliggende belangen van de onderhandelaars gekozen moeten worden als criteria, vooral als de voorkeuren betreffende de onderhandelingskwesties niet onafhankelijk van elkaar zijn. We laten zien dat het gebruik van belangen als criteria flexibeler is dan het modelleren van voorwaardelijke voorkeuren, en een betere uitleg biedt van de afgeleide voorkeuren.

- We introduceren een algemeen framework voor de representatie van kwalitatieve voorkeuren die gebaseerd zijn op meerdere criteria, genaamd Qualitative Preference Systems (QPS). Dit framework definieert uitkomsten als waardetoekenningen aan een verzameling variabelen met willekeurige domeinen, bevat een kennisbank die (harde) beperkingen op kan leggen en nieuwe (abstracte) concepten kan definiëren, en definieert drie typen criteria die gecombineerd kunnen worden in een boomstructuur. We laten zien dat het QPS-framework expressief is, aangezien het voorwaardelijke voorkeuren en onderliggende belangen kan representeren, alsook doelgebaseerde voorkeuren, bipolaire voorkeuren, en voorkeuren die gerepresenteerd zijn in twee andere bekende frameworks die representatief zijn voor een groot aantal puur kwalitatieve voorkeursrepresentatieframeworks. Bovendien laten we zien dat de doelgebaseerde variant van QPS net zo expressief is.

- Voor alle voorgestelde voorkeursrepresentatieframeworks definiëren we bijbehorende argumentatieframeworks, die een logische taal, een verzameling inferentieregels en een overwinningsrelatie bevatten. Enkele van de argumentatieframeworks bieden ook de mogelijkheid om te redeneren met achtergrondkennis om informatie over de waardes van variabelen af te leiden.

- We introduceren een mechanisme om uitleg te genereren voor voorkeuren die gerepresenteerd zijn in een QPS. We gebruiken de intuïtie dat voorkeuren uitgelegd kunnen worden aan de hand van de criteria die doorslaggevend zijn in de globale voorkeur. Bovendien laten we zien hoe een systeem een door de gebruiker gegeven uitleg kan gebruiken om zijn huidige voorkeursmodel bij te werken.

- Tenslotte introduceren we een modale logica, genaamd Multi-Attribute Preference Logic (MPL), die een taal biedt om verscheidene strategieën uit te drukken die voorkeuren tussen objecten kwalitatief afleiden uit een rangorde van eigenschappen. Er zijn drie van zulke strategieën uit de literatuur over doelgebaseerde voorkeuren gemodelleerd. De toegevoegde waarde van de logica is de mogelijkheid om niet alleen te redeneren over welke objecten geprefereerd worden volgens een bepaalde ordening, maar ook over de relaties tussen verschillende ordeningen.

Curriculum vitae

Wietske Visser was born in Roden, The Netherlands, on 13 July 1984. She completed her secondary education at the Willem Lodewijk Gymnasium in Groningen in 2002. In 2005 she obtained a BA degree in Linguistics (cum laude) from Leiden University. After that she obtained a MSc degree in Cognitive Artificial Intelligence (cum laude) from Utrecht University in 2008. Since 2008 she has worked at Delft University of Technology as a PhD researcher.

Dankwoord

Dit proefschrift markeert het einde van ruim vier jaar als promovendus. Deze periode heeft, naast dit proefschrift, een schat aan ervaringen opgeleverd. Op deze plek wil ik de mensen bedanken die hierbij een belangrijke rol hebben gespeeld.

Allereerst wil ik mijn afstudeerbegeleiders in Utrecht, Henry Prakken en Gerard Vreeswijk, bedanken voor hun aanmoediging om aan een promotietraject te beginnen. Ook al ben ik daarvoor uiteindelijk in Delft terechtgekomen, het is jullie vertrouwen dat me heeft gestimuleerd om een plek als promovendus te zoeken.

Dan zijn er natuurlijk mijn promotor en copromotor, Catholijn en Koen. Onder jullie begeleiding heb ik veel geleerd. Koen, bedankt voor je altijd relevante feedback op mijn werk. Catholijn, bedankt voor je steun toen ik het minder zag zitten.

Er zijn twee mensen met wie ik bijna de gehele afgelopen vier jaar mijn werkkamer heb gedeeld. Joost en Alina, bedankt voor alle meer of minder nuttige, maar altijd interessante discussies.

Bedankt ook, mijn academische broertjes en zusjes: Alina, Dmytro, Tim en Iris, voor het delen van ervaringen, frustraties en hoogtepunten. Het is goed te weten dat je niet alleen staat. Jullie en Nike, Zhenke, Hani, Hantao, Chris, Reyhan en Maaike, bedankt voor alle activiteiten die het leven van een promovendus leuk maken: colaquia, lunchen bij Ikea, badmintonnen, mahjong, karaoke en dumplings, barbecueën in de Delftse Hout, tripjes naar België, Texel, Drenthe en Indonesië, of gewoon een kletspraatje tussendoor. Dankzij jullie heb ik me altijd thuis gevoeld bij MMI/II.

Daarnaast wil ik ook mijn vrienden van buiten II bedanken voor de nodige afleiding, en alle Krashnezen voor de geweldige tijd bij Krashna.

Tot slot wil ik mijn ouders, Tom en Hetty, mijn zusjes, Hanne en Jeltje, en Bernard bedanken. Jullie vormen het thuisfront waar ik me altijd gesteund en geliefd weet.

Bibliography

[1] Leila Amgoud, Jean-François Bonnefon, and Henri Prade. An argumentation-based approach to multiple criteria decision. In *8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, pages 269–280, 2005.

[2] Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29:125–169, 2002.

[3] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraïtis. A unified and general framework for argumentation-based negotiation. In *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 967–974, 2007.

[4] Leila Amgoud, Nicolas Maudet, and Simon Parsons. Modelling dialogues using argumentation. In *4th International Conference on MultiAgent Systems*, pages 31–38, 2000.

[5] Leila Amgoud, Simon Parsons, and Nicolas Maudet. Arguments, dialogue, and negotiation. In *14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342, 2000.

[6] Leila Amgoud and Henri Prade. Reaching agreement through argumentation: A possibilistic approach. In *9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 175–182, 2004.

[7] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.

[8] Hajnal Andréka, Mark Ryan, and Pierre-Yves Schobbens. Operators and laws for combining preference relations. *Journal of Logic and Computation*, 12(1):13–53, 2002.

[9] Katie Atkinson, Trevor J. M. Bench-Capon, and Sanjay Modgil. Argumentation for decision support. In *17th International Conference on Database and Expert Systems Applications (DEXA 2006)*, pages 822–831, 2006.

[10] Reyhan Aydoğan and Pınar Yolum. Heuristics for CP-nets: Negotiating effectively with partial preferences. In *2nd International Working Conference on Human Factors and Computational Models in Negotiation (HuCom 2010)*, 2010.

[11] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen V. Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn M. Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, Accepted.

[12] Tim Baarslag, Koen V. Hindriks, and Catholijn M. Jonker. Acceptance conditions in automated negotiation. In *4th International Workshop on Agent-based Complex Automated Negotiations (ACAN 2011)*, 2011.

[13] Tim Baarslag, Koen V. Hindriks, and Catholijn M. Jonker. Towards a quantitative concession-based classification method of negotiation strategies. In *Agents in Principle, Agents in Practice. Lecture Notes of The 14th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA-2011)*, pages 143–158. 2011.

[14] Tim Baarslag, Koen V. Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The first Automated Negotiating Agents Competition (ANAC 2010). In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations*, Studies in Computational Intelligence, pages 113–135. Springer, 2012.

[15] Fahiem Bacchus and Adam J. Grove. Graphical models for preference and utility. In *11th Annual Conference on Uncertainty in Artificial Intelligence (UAI 1995)*, pages 3–10, 1995.

[16] Trevor Bench-Capon and Katie Atkinson. Abstract argumentation and values. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 45–64. Springer, 2009.

[17] Trevor J. M. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[18] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.

[19] Salem Benferhat, Didier Dubois, Souhila Kaci, and Henri Prade. Bipolar possibility theory in preference modeling: Representation, fusion and optimal solutions. *Information Fusion*, 7(1):135–150, 2006.

[20] Salem Benferhat and Karima Sedki. Two alternatives for handling preferences in qualitative choice logic. *Fuzzy Sets and Systems*, 159(15):1889–1912, 2008.

[21] Jamal Bentahar and Jihad Labban. An argumentation-driven model for flexible and efficient persuasive negotiation. *Group Decision and Negotiation*, 20(4):411–435, 2011.

[22] Johan van Benthem, Patrick Girard, and Olivier Roy. Everything else being equal: A modal logic for ceteris paribus preferences. *Journal of Philosophical Logic*, 38:83–125, 2009.

[23] Patrick Blackburn and Jerry Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

[24] Blai Bonet and Hector Geffner. Arguing for decisions: A qualitative model of decision making. In *12th Conference on Uncertainty in Artificial Intelligence (UAI 1996)*, pages 98–105, 1996.

[25] Jean-François Bonnefon, Didier Dubois, Hélène Fargier, and Sylvie Leblois. Qualitative heuristics for balancing the pros and cons. *Theory and Decision*, 65(1):71–95, 2008.

[26] Jean-François Bonnefon and Hélène Fargier. Comparing sets of positive and negative arguments: Empirical assessment of seven qualitative rules. In *17th European Conference on Artificial Intelligence (ECAI 2006)*, pages 16–20, 2006.

[27] Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera. Learning conditionally lexicographic preference relations. In *19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 269–274, 2010.

[28] Craig Boutilier. Toward a logic for qualitative decision theory. In *4th International Conference on Principles of Knowledge Representation and Reasoning (KR 1994)*, pages 75–86, 1994.

[29] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[30] Ronen I. Brafman and Carmel Domshlak. Introducing variable importance tradeoffs into CP-nets. In *18th Conference on Uncertainty in Artificial Intelligence (UAI 2002)*, pages 69–76, 2002.

[31] Ronen I. Brafman and Carmel Domshlak. Preference handling: An introductory tutorial. *AI Magazine*, 30(1):58–86, 2009.

[32] Ronen I. Brafman, Carmel Domshlak, and Solomon E. Shimony. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research*, 25(1):389–424, 2006.

[33] Gerhard Brewka. A rank based description language for qualitative preferences. In *16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 303–307, 2004.

[34] Gerhard Brewka, Salem Benferhat, and Daniel Le Berre. Qualitative choice logic. *Artificial Intelligence*, 157(1-2):203–237, 2004.

[35] Willem-Paul Brinkman, Joost Broekens, Catholijn M. Jonker, and John-Jules Ch. Meyer. Getting a grip on emotions in negotiations: The possibilities of ICT. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009)*, pages 345–348, 2009.

[36] Joost Broekens, Maaike Harbers, Willem-Paul Brinkman, Karel van den Bosch, Catholijn M. Jonker, and John-Jules Meyer. Virtual reality negotiation training increases negotiation knowledge and skill. In *12th International Conference on Intelligent Virtual Agents (IVA 2012)*, 2012.

[37] Joost Broekens, Maaike Harbers, Willem-Paul Brinkman, Catholijn Jonker, Karel van den Bosch, and John-Jules Meyer. Validity of a virtual negotiation training. In *10th International Conference on Intelligent Virtual Agents (IVA 2011)*, pages 435–436, 2011.

[38] Joost Broekens, Maaike Harbers, Koen Hindriks, Karel van den Bosch, Catholijn Jonker, and John-Jules Meyer. Do you get it? User-evaluated explainable BDI agents. In Jürgen Dix and Cees Witteveen, editors, *Multiagent System Technologies*, volume 6251 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2010.

[39] Joost Broekens, Catholijn M. Jonker, and John-Jules Ch. Meyer. Affective negotiation support systems. *Journal of Ambient Intelligence and Smart Environments*, 2(2):121–144, 2010.

[40] Giuseppe Carenini and Johanna D. Moore. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–952, 2006.

[41] Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Expressive power of weighted propositional formulas for cardinal preference modelling. In *10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, volume 145-152, 2006.

[42] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Magazine*, Winter:37–46, 2008.

[43] Yann Chevaleyre, Frédéric Koriche, Jérôme Lang, Jérôme Mengin, and Bruno Zanuttini. Learning ordinal preferences on multiattribute domains: The case of CP-nets. In Johannes Fürnkranz and Eyke Hüllermeier, editors, *Preference Learning*, pages 273–296. Springer, 2011.

[44] Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Expressive power and succinctness of propositional languages for preference representation. In *9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 203–212, 2004.

[45] Anja Dieckmann, Katrin Dippold, and Holger Dietrich. Compensatory versus noncompensatory models for predicting consumer preferences. *Judgment and Decision Making*, 4(3):200–213, 2009.

[46] Yannis Dimopoulos, Pavlos Moraïtis, and Alexis Tsoukiàs. Qualitative preference modelling in constraint satisfaction. In G. Della Riccia, D. Dubois, R. Kruse, and H.-J. Lenz, editors, *Decision Theory and Multi-Agent Planning*, pages 15 – 29. Springer, 2004.

[47] Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in AI: An overview. *Artificial Intelligence*, 175(7-8):1037–1052, 2011.

[48] Carmel Domshlak, Steve Prestwich, Francesca Rossi, K. Brent Venable, and Toby Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *Journal of Heuristics*, 12(4):263–285, 2006.

[49] Jon Doyle and Richmond H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68, 1999.

[50] Didier Dubois, Hélène Fargier, and Jean-François Bonnefon. On the qualitative comparison of decisions having positive and negative features. *Journal of Artificial Intelligence Research*, 32:385–417, 2008.

[51] Didier Dubois, Hélène Fargier, and Patrice Perny. Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach. *Artificial Intelligence*, 148:219–260, 2003.

[52] Phan Minh 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.

[53] Peyman Faratin, Carles Sierra, and Nick R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

[54] Peter C. Fishburn. *Utility Theory for Decision Making*. Wiley & Sons, 1969.

[55] Ya'akov Gal, Sohan D'souza, Phillippe Pasquier, Iyad Rahwan, and Sherrief Abdallah. The effects of goal revelation on computer-mediated negotiation. In *Annual meeting of the Cognitive Science Society (CogSci 2009)*, pages 2614–2619, 2009.

[56] Romain Gérard, Souhila Kaci, and Henri Prade. Ranking alternatives on the basis of generic constraints and examples: a possibilistic approach. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 393–398, 2007.

[57] Gerd Gigerenzer, Peter M. Todd, and ABC Group. *Simple heuristics that make us smart*. Oxford University Press, 1999.

[58] Patrick Girard. *Modal Logic for Belief and Preference Change*. PhD thesis, Stanford University, 2008.

[59] Christophe Gonzales and Patrice Perny. GAI networks for utility elicitation. In *9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 224–234, 2004.

[60] Joseph Y. Halpern and Gerhard Lakemeyer. Multi-agent only knowing. *Journal of Logic and Computation*, 11(1):41–70, 2001.

[61] Sven Ove Hansson. Preference logic. In D.M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 319–393. Kluwer, 2nd edition, 2001.

[62] Frederick Herzberg, Bernard Mausner, and Barbara Bloch Snyderman. *The motivation to work*. Wiley, 1959.

[63] Koen Hindriks, Catholijn Jonker, and Wietske Visser. Reasoning about multi-attribute preferences. In *21st Benelux Conference on Artificial Intelligence (BNAIC 2009)*, pages 319–320, 2009.

[64] Koen Hindriks, Catholijn Jonker, and Wietske Visser. Reasoning about multi-attribute preferences (extended abstract). In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1147–1148, 2009.

[65] Koen V. Hindriks, Wietske Visser, and Catholijn M. Jonker. Multi-attribute preference logic. In N. Desai, A. Liu, and M. Winikoff, editors, *PRIMA 2010*, volume 7057 of *Lecture Notes in Artificial Intelligence*, pages 181–195. 2012.

[66] Franz Huber. Belief and degrees of belief. In Franz Huber and Christoph Schmidt-Petri, editors, *Degrees of Belief*, volume 342 of *Synthese Library*, pages 1–33. Springer, 2009.

[67] Nick R. Jennings, Simon Parsons, Pablo Noriega, and Carles Sierra. On argumentation-based negotiation. In *International Workshop on Multi-Agent Systems (IWMAS 1998)*, 1998.

[68] Catholijn M. Jonker. The pocket negotiator, synergy between man and machine. Grant proposal, 2007.

[69] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15:221–252, 2007.

[70] Souhila Kaci. *Working with Preferences: Less Is More*. Springer, 2011.

[71] Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48(3):730–751, 2008.

[72] Antonis Kakas and Pavlos Moraïtis. Argumentation based decision making for autonomous agents. In *2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 883–890, 2003.

[73] Nishan C. Karunatillake, Nicholas R. Jennings, Iyad Rahwan, and Timothy J. Norman. Argument-based negotiation in a social context. In *2nd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2005)*, pages 104–121, 2005.

[74] Ralph L. Keeney. Analysis of preference dependencies among objectives. *Operations Research*, 29(6):1105–1120, 1981.

[75] Ralph L. Keeney. *Value-Focused Thinking: A Path to Creative Decisionmaking*. Harvard University Press, 1992.

[76] Ralph L. Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, 1993.

[77] Iris van de Kieft, Catholijn M. Jonker, and M. Birna van Riemsdijk. Explaining negotiation: Obtaining a shared mental model of preferences. In *Modern Approaches in Applied Intelligence*, volume 6704 of *Lecture Notes in Computer Science*, pages 120–129. Springer, 2011.

[78] Iris van de Kieft, Catholijn M. Jonker, and M. Birna van Riemsdijk. Improving user and decision support system teamwork: An approach based on shared mental models. In *6th International Workshop on Explanation-aware Computing (ExaCt 2011)*, pages 61–70, 2011.

[79] Iris van de Kieft, Catholijn M. Jonker, and M. Birna van Riemsdijk. Shared mental models for decision support systems and their users. In *3rd International Workshop on Collaborative Agents - REsearch and development (CARE 2011)*, pages 54–63, 2011.

[80] David A. Klein and Edward H. Shortliffe. A framework for explaining decision-theoretic advice. *Artificial Intelligence*, 67(2):201–243, 1994.

[81] Sarit Kraus, Katia Sycara, and Amir Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104:1–69, 1998.

[82] Christophe Labreuche. A general framework for explaining the results of a multi-attribute preference model. *Artificial Intelligence*, 175(7-8):1410–1448, 2011.

[83] Jérôme Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.

[84] Jérôme Lang. Vote and aggregation in combinatorial domains with structured preferences. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 1366–1371, 2007.

[85] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. An efficient procedure for collective decision-making with CP-nets. In *19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 375–380, 2010.

[86] Fenrong Liu. *Changing for the Better: Preference Dynamics and Agent Diversity*. PhD thesis, Universiteit van Amsterdam, 2008.

[87] Nicolas Maudet, Simon Parsons, and Iyad Rahwan. Argumentation in multi-agent systems: Context and recent developments. In *Argumentation in Multi-Agent Systems*, volume 4766 of *Lecture Notes in Computer Science*, pages 1–16. 2007.

[88] Mohamed Mbarki, Jamal Bentahar, Bernard Moulin, and Ahmad Moazin. Argumentation-based negotiation using constraints: Specification and implementation. In *2nd International Working Conference on Human Factors and Computational Models in Negotiation (HuCom 2010)*, 2010.

[89] Peter McBurney, Rogier M. van Eijk, Simon Parsons, and Leila Amgoud. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

[90] Peter McBurney, Simon Parsons, and Michael Wooldridge. Desiderata for agent argumentation protocols. In *1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 402–409, 2002.

[91] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. Experiments in dynamic critiquing. In *10th international Conference on Intelligent User Interfaces (IUI 2005)*, pages 175–182, 2005.

[92] Bernard Moulin, Hengameh Irandoust, Micheline Bélanger, and Gaëlle Desbordes. Explanation and argumentation capabilities: Towards the creation of more persuasive agents. *Artificial Intelligence Review*, 17(3):169–222, 2002.

[93] Wassila Ouerdane, Nicolas Maudet, and Alexis Tsoukiàs. Argument schemes and critical questions for decision aiding process. In *2nd Conference on Computational Models of Argument (COMMA 2008)*, pages 285–296, 2008.

[94] Wassila Ouerdane, Nicolas Maudet, and Alexis Tsoukiàs. Argumentation theory and decision aiding. In Matthias Ehrgott, José Rui Figueira, and Salvatore Greco, editors, *New Trends in Multiple Criteria Decision Analysis*. Springer, 2010.

[95] Meltem Öztürk, Alexis Tsoukiàs, and Philippe Vincke. Preference modelling. In José Figueira, Salvatore Greco, and Matthias Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 27–59. 2005.

[96] Simon Parsons and N. R. Jennings. Negotiation through argumentation - a preliminary report. In *2nd International Conference on Multiagent Systems (ICMAS 1996)*, pages 267–274, 1996.

[97] Simon Parsons, Carles Sierra, and Nick Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[98] Philippe Pasquier, Ramon Hollands, Iyad Rahwan, Frank Dignum, and Liz Sonenberg. An empirical study of interest-based negotiation. *Autonomous Agents and Multi-Agent Systems*, pages 1–40, 2010.

[99] John L. Pollock. Defeasible reasoning with variable degrees of justification. *Artificial Intelligence*, 133(1-2):233–282, 2001.

[100] Alina Pommeranz, Joost Broekens, Pascal Wiggers, Willem-Paul Brinkman, and Catholijn Jonker. Designing interfaces for explicit preference elicitation: a user-centered investigation of preference representation and elicitation process. *User Modeling and User-Adapted Interaction*, 22(4-5):357–397, 2012.

[101] Alina Pommeranz, Christian Detweiler, Pascal Wiggers, and Catholijn Jonker. Elicitation of situated values: need for tools to help stakeholders and designers to reflect and communicate. *Ethics and Information Technology*, Online First, 2011.

[102] Alina Pommeranz, Pascal Wiggers, Willem-Paul Brinkman, and Catholijn M. Jonker. Social acceptance of negotiation support systems: scenario-based exploration with focus groups and online survey. *Cognition, Technology & Work*, Online First, 2011.

[103] Henry Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *10th International Conference on Artificial Intelligence and Law (ICAIL 2005)*, pages 85–94, 2005.

[104] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[105] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.

[106] Iyad Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, University of Melbourne, 2004.

[107] Iyad Rahwan. Guest editorial: Argumentation in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 11(2):115–125, 2005.

[108] Iyad Rahwan, Philippe Pasquier, Liz Sonenberg, and Frank Dignum. On the benefits of exploiting underlying goals in argument-based negotiation. In *22nd Conference on Artificial Intelligence (AAAI 2007)*, pages 116–121, 2007.

[109] Iyad Rahwan, Philippe Pasquier, Liz Sonenberg, and Frank Dignum. A formal analysis of interest-based negotiation. *Annals of Mathematics and Artificial Intelligence*, 55(3):253–276, 2009.

[110] Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2004.

[111] Iyad Rahwan, Liz Sonenberg, and Frank P. M. Dignum. On interest-based negotiation. In *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Computer Science*, pages 383–401. 2004.

[112] Sarvapali D. Ramchurn, Nicholas R. Jennings, and Carles Sierra. Persuasive negotiation for autonomous agents: A rhetorical approach. In *3rd Workshop on Computational Models of Natural Argument (CMNA 2003)*, 2003.

[113] Jeffrey Rosenschein and Gilad Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers*. MIT Press, 1994.

[114] Carles Sierra, Nick R. Jennings, Pablo Noriega, and Simon Parsons. A framework for argumentation-based negotiation. In *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, volume 1365 of *Lecture Notes in Computer Science*, pages 177–192. 1997.

[115] Markus Stolze and Michael Ströbel. Dealing with learning in ecommerce product navigation and decision support: The teaching salesman problem. In *2nd Interdisciplinary World Congress on Mass Customization and Personalization (MCPC 2003)*, 2003.

[116] Katia P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.

[117] Dmytro Tykhonov. *Designing Generic and Efficient Negotiation Strategies*. PhD thesis, Delft University of Technology, 2010.

[118] Joel Uckelman. *More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains*. PhD thesis, University of Amsterdam, 2009.

[119] Joel Uckelman, Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Representing utility functions via weighted goals. *Mathematical Logic Quarterly*, 55(4):341–361, 2009.

[120] Paolo Viappiani, Pearl Pu, and Boi Faltings. Preference-based search with adaptive recommendations. *AI Communications*, 21(2):155–175, 2008.

[121] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. A framework for qualitative multi-criteria preferences. In *4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, pages 243–248, 2012.

[122] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. A framework for qualitative multi-criteria preferences: Extended abstract. In *24th Benelux Conference on Artificial Intelligence (BNAIC 2012)*, 2012.

[123] Wietske Visser, Reyhan Aydoğan, Koen V. Hindriks, and Catholijn M. Jonker. Qualitative Preference Systems: A framework for qualitative multi-criteria preferences. Submitted.

[124] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Argumentation-based preference modelling with incomplete information. In *CLIMA X*, volume 6214 of *Lecture Notes in Artificial Intelligence*, pages 141–157. 2010.

[125] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. An argumentation framework for deriving qualitative risk sensitive preferences. In *Modern Approaches in Applied Intelligence*, volume 6704 of *Lecture Notes in Computer Science*, pages 556–565, 2011.

[126] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Interest-based preference reasoning. In *3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, pages 79–88, 2011.

[127] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Argumentation-based qualitative preference modelling with incomplete and uncertain information. *Group Decision and Negotiation*, 21(1):99–127, 2012.

[128] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. An argumentation framework for qualitative multi-criteria preferences. In *Theory and Applications of Formal Argumentation (TAFA 2011)*, volume 7132 of *Lecture Notes in Artificial Intelligence*, pages 85–98. 2012.

[129] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Explaining qualitative preference models. In *6th Multidisciplinary Workshop on Advances in Preference Handling (M-PREF 2012)*, 2012.

[130] Wietske Visser, Koen V. Hindriks, and Catholijn M. Jonker. Goal-based qualitative preference systems. In *10th International Workshop on Declarative Agent Languages and Technologies (DALT 2012)*, 2012.

[131] Gerard A. W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1-2):225–279, 1997.

[132] Toby Walsh. Representing and reasoning with preferences. *AI Magazine*, 28(4):59–69, 2007.

[133] Tom van der Weide. *Arguing to Motivate Decisions*. PhD thesis, Universiteit Utrecht, 2011.

[134] Tom van der Weide, Frank Dignum, John-Jules Meyer, Henry Prakken, and Gerard Vreeswijk. Practical reasoning using values: Giving meaning to values. In *Argumentation in Multi-Agent Systems*, volume 6057 of *Lecture Notes in Computer Science*, pages 79–93. 2009.

[135] Michael P. Wellman and Jon Doyle. Preferential semantics for goals. In *9th National Conference on Artificial Intelligence (AAAI 1991)*, pages 698–703, 1991.

[136] Nic Wilson. Extending CP-nets with stronger conditional preference statements. In *19th National Conference on Artificial Intelligence (AAAI 2004)*, pages 735–741, 2004.

[137] Nic Wilson. Computational techniques for a simple theory of conditional preferences. *Artificial Intelligence*, 175(7-8):1053–1091, 2011.

[138] Georg Henrik von Wright. *The Logic of Preference: An Essay*. Edinburgh University Press, 1963.

SIKS Dissertatiereeks

2009

2009-01 Rasa Jurgelenaite (RUN), Symmetric Causal Independence Models
2009-02 Willem Robert van Hage (VU), Evaluating Ontology-Alignment Techniques
2009-03 Hans Stol (UvT), A Framework for Evidence-based Policy Making Using IT
2009-04 Josephine Nabukenya (RUN), Improving the Quality of Organisational Policy Making using Collaboration Engineering
2009-05 Sietse Overbeek (RUN), Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality
2009-06 Muhammad Subianto (UU), Understanding Classification
2009-07 Ronald Poppe (UT), Discriminative Vision-Based Recovery and Recognition of Human Motion
2009-08 Volker Nannen (VU), Evolutionary Agent-Based Policy Analysis in Dynamic Environments
2009-09 Benjamin Kanagwa (RUN), Design, Discovery and Construction of Service-oriented Systems
2009-10 Jan Wielemaker (UVA), Logic programming for knowledge-intensive interactive applications
2009-11 Alexander Boer (UVA), Legal Theory, Sources of Law & the Semantic Web
2009-12 Peter Massuthe (TUE, Humboldt-Universität zu Berlin), Operating Guidelines for Services
2009-13 Steven de Jong (UM), Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU), From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)
2009-15 Rinke Hoekstra (UVA), Ontology Representation - Design Patterns and Ontologies that Make Sense
2009-16 Fritz Reul (UvT), New Architectures in Computer Chess
2009-17 Laurens van der Maaten (UvT), Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI), Armada, An Evolving Database System
2009-19 Valentin Robu (CWI), Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets
2009-20 Bob van der Vecht (UU), Adjustable Autonomy: Controling Influences on Decision Making
2009-21 Stijn Vanderlooy (UM), Ranking and Reliable Classification
2009-22 Pavel Serdyukov (UT), Search For Expertise: Going beyond direct evidence
2009-23 Peter Hofgesang (VU), Modelling Web Usage in a Changing Environment
2009-24 Annerieke Heuvelink (VUA), Cognitive Models for Training Simulations
2009-25 Alex van Ballegooij (CWI), RAM: Array Database Management through Relational Mapping
2009-26 Fernando Koch (UU), An Agent-Based Model for the Development of Intelligent Mobile Services
2009-27 Christian Glahn (OU), Contextual Support of social Engagement and Reflection on the Web
2009-28 Sander Evers (UT), Sensor Data Management with Probabilistic Models
2009-29 Stanislav Pokraev (UT), Model-Driven Semantic Integration of Service-Oriented Applications
2009-30 Marcin Zukowski (CWI), Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA), A Closer Look at Learning Relations from Text
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), Architectural Knowledge Management: Supporting Architects and Auditors
2009-33 Khiet Truong (UT), How Does Real Affect Affect Affect Recognition In Speech?
2009-34 Inge van de Weerd (UU), Advancing in Software Product Management: An Incremental Method Engineering Approach
2009-35 Wouter Koelewijn (UL), Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
2009-36 Marco Kalz (OUN), Placement Support for Learners in Learning Networks
2009-37 Hendrik Drachsler (OUN), Navigation Support for Learners in Informal Learning Networks
2009-38 Riina Vuorikari (OU), Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
2009-39 Christian Stahl (TUE, Humboldt-Universität zu Berlin), Service Substitution - A Behavioral Approach Based on Petri Nets
2009-40 Stephan Raaijmakers (UvT), Multinomial Language Learning: Investigations into the Geometry of Language
2009-41 Igor Berezhnyy (UvT), Digital Analysis of Paintings

2009-42 Toine Bogers (UvT), Recommender Systems for Social Bookmarking
2009-43 Virginia Nunes Leal Franqueira (UT), Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
2009-44 Roberto Santana Tapia (UT), Assessing Business-IT Alignment in Networked Organizations
2009-45 Jilles Vreeken (UU), Making Pattern Mining Useful
2009-46 Loredana Afanasiev (UvA), Querying XML: Benchmarks and Recursion

2010

2010-01 Matthijs van Leeuwen (UU), Patterns that Matter
2010-02 Ingo Wassink (UT), Work flows in Life Science
2010-03 Joost Geurts (CWI), A Document Engineering Model and Processing Framework for Multimedia documents
2010-04 Olga Kulyk (UT), Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments
2010-05 Claudia Hauff (UT), Predicting the Effectiveness of Queries and Retrieval Systems
2010-06 Sander Bakkes (UvT), Rapid Adaptation of Video Game AI
2010-07 Wim Fikkert (UT), Gesture interaction at a Distance
2010-08 Krzysztof Siewicz (UL), Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments
2010-09 Hugo Kielman (UL), A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging
2010-10 Rebecca Ong (UL), Mobile Communication and Protection of Children
2010-11 Adriaan Ter Mors (TUD), The world according to MARP: Multi-Agent Route Planning
2010-12 Susan van den Braak (UU), Sensemaking software for crime analysis
2010-13 Gianluigi Folino (RUN), High Performance Data Mining using Bio-inspired techniques
2010-14 Sander van Splunter (VU), Automated Web Service Reconfiguration
2010-15 Lianne Bodenstaff (UT), Managing Dependency Relations in Inter-Organizational Models
2010-16 Sicco Verwer (TUD), Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU), Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
2010-18 Charlotte Gerritsen (VU), Caught in the Act: Investigating Crime by Agent-Based Simulation
2010-19 Henriette Cramer (UvA), People's Responses to Autonomous and Adaptive Systems
2010-20 Ivo Swartjes (UT), Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
2010-21 Harold van Heerde (UT), Privacy-aware data management by means of data degradation
2010-22 Michiel Hildebrand (CWI), End-user Support for Access to Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU), The Logical Structure of Emotions
2010-24 Dmytro Tykhonov (TUD), Designing Generic and Efficient Negotiation Strategies
2010-25 Zulfiqar Ali Memon (VU), Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective
2010-26 Ying Zhang (CWI), XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines
2010-27 Marten Voulon (UL), Automatisch contracteren
2010-28 Arne Koopman (UU), Characteristic Relational Patterns
2010-29 Stratos Idreos(CWI), Database Cracking: Towards Auto-tuning Database Kernels
2010-30 Marieke van Erp (UvT), Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval
2010-31 Victor de Boer (UVA), Ontology Enrichment from Heterogeneous Sources on the Web
2010-32 Marcel Hiel (UvT), An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems
2010-33 Robin Aly (UT), Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval
2010-34 Teduh Dirgahayu (UT), Interaction Design in Service Compositions
2010-35 Dolf Trieschnigg (UT), Proof of Concept: Concept-based Biomedical Information Retrieval
2010-36 Jose Janssen (OU), Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification
2010-37 Niels Lohmann (TUE), Correctness of services and their composition
2010-38 Dirk Fahland (TUE), From Scenarios to components
2010-39 Ghazanfar Farooq Siddiqui (VU), Integrative modeling of emotions in virtual agents
2010-40 Mark van Assem (VU), Converting and Integrating Vocabularies for the Semantic Web
2010-41 Guillaume Chaslot (UM), Monte-Carlo Tree Search
2010-42 Sybren de Kinderen (VU), Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach
2010-43 Peter van Kranenburg (UU), A Computational Approach to Content-Based Retrieval of Folk Song Melodies
2010-44 Pieter Bellekens (TUE), An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain
2010-45 Vasilios Andrikopoulos (UvT), A theory and model for the evolution of software services
2010-46 Vincent Pijpers (VU), e3alignment: Exploring Inter-Organizational Business-ICT Alignment
2010-47 Chen Li (UT), Mining Process Model Variants: Challenges, Techniques, Examples
2010-49 Jahn-Takeshi Saito (UM), Solving difficult game positions
2010-50 Bouke Huurnink (UVA), Search in Audiovisual Broadcast Archives
2010-51 Alia Khairia Amin (CWI), Understanding and supporting information seeking tasks in multiple sources
2010-52 Peter-Paul van Maanen (VU), Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

2010-53 Edgar Meij (UVA), Combining Concepts and Language Models for Information Access

2011

2011-01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models
2011-02 Nick Tinnemeier(UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
2011-03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems
2011-04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms
2011-05 Base van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline
2011-06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage
2011-07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction
2011-08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues
2011-09 Tim de Jong (OU), Contextualised Mobile Media for Learning
2011-10 Bart Bogaert (UvT), Cloud Content Contention
2011-11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective
2011-12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining
2011-13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling
2011-14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets
2011-15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
2011-16 Maarten Schadd (UM), Selective Search in Games of Different Complexity
2011-17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
2011-18 Mark Ponsen (UM), Strategic Decision-Making in complex games
2011-19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles
2011-20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
2011-21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
2011-22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
2011-23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
2011-24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
2011-25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
2011-26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
2011-27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
2011-28 Rianne Kaptein(UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
2011-29 Faisal Kamiran (TUE), Discrimination-aware Classification
2011-30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
2011-31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
2011-32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
2011-33 Tom van der Weide (UU), Arguing to Motivate Decisions
2011-34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
2011-35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
2011-36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
2011-37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
2011-38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization
2011-39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
2011-40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
2011-41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
2011-42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
2011-43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
2011-44 Boris Reuderink (UT), Robust Brain-Computer Interfaces
2011-45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
2011-46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
2011-47 Azizi Bin Ab Aziz(VU), Exploring Computational Models for Intelligent Support of Persons with Depression
2011-48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
2011-49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012

2012-01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
2012-02 Muhammad Umair(VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
2012-03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
2012-04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
2012-06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
2012-07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
2012-08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
2012-09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
2012-10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
2012-11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
2012-12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
2012-13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
2012-14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
2012-15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes
2012-16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
2012-17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
2012-18 Eltjo Poort (VU), Improving Solution Architecting Practices
2012-19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
2012-20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
2012-21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
2012-22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
2012-23 Christian Mühl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
2012-24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
2012-25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
2012-26 Emile de Maat (UVA), Making Sense of Legal Text
2012-27 Hayrettin Gürkök (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
2012-28 Nancy Pascall (UvT), Engendering Technology Empowering Women
2012-29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
2012-30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
2012-31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure