The Local Geometry of Multiattribute Tradeoff Preferences

by

Michael McGeachie

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

Author ....
Department of Electrical Engineering and Computer Science
February 2nd, 2007

Certified by ....
Jon Doyle
SAS Institute Distinguished Professor of Computer Science,
North Carolina State University
Thesis Supervisor

Certified by ....
Peter Szolovits
Professor of Computer Science and Engineering,
Massachusetts Institute of Technology
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

The Local Geometry of Multiattribute Tradeoff Preferences

by

Michael McGeachie

Submitted to the Department of Electrical Engineering and Computer Science
on February 2nd, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Existing preference reasoning systems have been successful in simple domains. Broader success requires more natural and more expressive preference representations. This thesis develops a representation of logical preferences that combines numerical tradeoff ratios between partial outcome descriptions with qualitative preference information. We argue our system is unique among preference reasoning systems; previous work has focused on qualitative or quantitative preferences, tradeoffs, exceptions and generalizations, or utility independence, but none have combined all of these expressions under a unified methodology.

We present new techniques for representing and giving meaning to quantitative tradeoff statements between different outcomes. The tradeoffs we consider can be multi-attribute tradeoffs relating more than one attribute at a time, they can refer to discrete or continuous domains, be conditional or unconditional, and quantified or qualitative. We present related methods of representing judgments of attribute importance. We then build upon a methodology for representing arbitrary qualitative *ceteris paribus* preference, or preferences "other things being equal," as presented in [MD04]. Tradeoff preferences in our representation are interpreted as constraints on the partial derivatives of the utility function. For example, a decision maker could state that "Color is five times as important as price, availability, and time," a sentiment one might express in the context of repainting a home, and this is interpreted as indicating that utility increases in the positive color direction five times faster than utility increases in the positive price direction. We show that these representations generalize both the economic notion of marginal rates of substitution and previous representations of preferences in AI.

Thesis Supervisor: Jon Doyle
Title: SAS Institute Distinguished Professor of Computer Science,
North Carolina State University

Thesis Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering,
Massachusetts Institute of Technology

Acknowledgments

It is my continuing privilege to work under the supervision of Professor Jon Doyle. His knowledge, scholarship, and willingness to continue working with me from separate universities has been my greatest professional asset in this work.

To my thesis committee members, Peter Szolovits and Patrick Winston, I owe thanks and appreciation. They have provided contrasting opinions and entertained my own professional idiosyncrasies. I am, to this day, amazed and grateful that Professor Szolovits, as the head of the Clinical Decision Making Group, has been so supportive of my work for so long.

Like any work this storied, this thesis has gone through many incarnations and phases, and in those past it benefitted greatly from the anonymous reviewers at various conferences for their extremely helpful comments and criticisms. Nameless ones, your help is noted, and duly received.

My family's confidence is both an irreplaceable and intrinsic part of anything I do. For everything, I thank my parents Robert and Catherine, and my brothers Patrick and Daniel. They have enabled this work to continue where it would not have otherwise.

I would be grateful for any small presence of Vanessa Lord's singular spirit in my life. That she has donated so much of it to me has been my greatest joy.

MIT has been a strange place to live and work during the years of this thesis. Some of the best memories of it are due to Michael Oltmans and the AI Lab's GSB club, and the cultural cohesion they both provide. It has been unforgettable.

And finally, without the support of DARPA, a training grant from the National Library of Medicine, and the Pfizer Corporation, none of this work would have been possible.

Contents

Chapter 1

Building Utility Functions from Diverse Tradeoffs and *Ceteris Paribus*Preferences

Knowledge of a user's preferences can be used to make decisions on behalf of the user. Direct and complete elicitation of preferences in the form of utility functions has enabled decision analysts to solve decision problems for users since von Neumann and Morgenstern [vNM44]. However, any preference information is valuable, even if it is somewhat vague or covers the domain only loosely.

This thesis presents methods to represent many different types of preferences and methods to compile utility functions from these preferences that can then be used to evaluate outcomes in accordance with the preferences. We present new techniques for representing and giving meaning to quantitative tradeoff statements between different outcomes; these are statements that compare two partial descriptions of outcomes, and are interpreted as stating that outcomes matching the first description are somehow "better" than the outcomes matching the second description. The tradeoffs we consider can be multi-attribute tradeoffs relating more than one attribute at a time, they can refer to discrete or continuous domains, be conditional or unconditional, and quantified or qualitative. We present related methods of representing judgements of attribute importance, a type of preference that figures prominently in applied decision analysis, and indicates that the weight given to some attribute or attributes in a decision should be greater than that given to other attributes. We then build upon a methodology for representing arbitrary qualitative *ceteris paribus* preference, preferences other things being equal, proposed in [MD04].

Next we consider how to create a utility function that represents a collection of tradeoff, importance, and *ceteris paribus*preferences. A utility function evaluates outcomes, giving higher values to outcomes that are more preferred. We say the utility function represents a set of preferences if the utility function gives an outcome a greater value than a second outcome whenever the first is preferred to the second by sound inference procedures on the preferences. We perform this task making a minimum of assumptions and restrictions on the form or character of the input preferences.

Our approach is in contrast to those of traditional decision analysis. Prior to formalization, traditional decision analysts identify the dimensions of a decision, assess variable independence, and elicit utility functions for each independent set of dimensions, frequently by posing variations on standard gambles to the user. We extend traditional decision analytic techniques to permit tentative formalizations to begin much earlier, and to work with much less information. These extensions are useful when the decision maker engages in protracted incremental deliberation, is unaware of the basic framing of a problem, or for one reason or another does not require the full machinery of traditional decision analysis. Specifically, we present representations for a variety of tradeoff preferences between concrete alternatives, importance statements relating the relative value of attributes, and qualitative *ceteris paribus*preferences.

Suppose Mike is using his personal online shopping agent to watch for deals on computer hardware he may find attractive. Mike will buy a new laptop if there is a good deal on one he likes. The agent retrieves a list of laptops for sale at various vendors' websites. Seeing the list, Mike decides that, with respect to price and speed, a $1500, 3 Ghz machine is preferable to a $2000, 3.4 Ghz machine, other things being equal. The agent then filters out some of the results that are too expensive. Thinking about it a little more, Mike decides that the first machine is much better than the other one, and decides that it is five times better. Mike isn't thinking too carefully about the relative weights that might be assigned to different dimensions in a hypothetical utility function, he just knows that one product is much more attractive than the other. Looking at some of the expensive options with many attractive features, Mike then realizes that adding features and adding ounces of weight at the same time is not what he wants. Mike tells the agent that *Weight* is more important than *Price*. The agent readjusts its evaluation of options, and shows more laptops ordered by weight, with several attractive light-weight options at the top of the list. Mike sees that there are some good machines available that are light, moderately powerful, and within his price range, but realizes that he must decide how big a screen and what resolution he needs to do his work on the road, since this adversely impacts the price and the weight. Mike decides a 12" screen on a 4.5 pound machine for $1700 is better than a 14" screen on a 6 pound machine for $1800. This suffices to order the remaining options in a way that suits Mike's intuitions, and ends up earmarking a machine for later consideration and probable purchase.

In traditional decision theory literature, starting with [vNM44] and continuing through [Sav54], [Fis64], and [KR76], there is a methodological assumption that the decision analyst will first interview the decision maker about what dimensions or attributes of the decision are of consequence. Then the decision analyst assesses utility functions on each of these dimensions by means of standard gambles. This requires the decision maker to think carefully about the upper and lower bounds of each dimension, consider his or her attitude toward risky hypothetical choices, and determine which attributes are utility independent of other attributes. Next the relative importance of each dimension must be assessed. These steps can be lengthy and time-consuming, and not necessary in every application.

Mike's interaction with his shopping agent typifies the applications we address. We imagine the decision maker might interact with a simple program that allows him or her to specify preferences in a variety of forms. Or perhaps a learning agent watches the user make other choices and decisions and slowly builds up a repertoire of past preferences. The user, the learning agent, or some intermediary process can then record qualitative *ceteris paribus*preferences involving multiple variables, for as many or as few variables and statements as is appropriate. Other preferences can be represented as bounds on quantitative tradeoffs between groups of variables or between "market baskets" of particular values of groups of variables. Our methodology does not require any statements of explicit utility independence, as this can frequently be inferred from the other preference statements. In general this preference collection phase is free of assumptions about the structure of the user's utility function.

We present tradeoff preferences that can be either quantitative or qualitative. I may feel that a reduction in one hour of travel time or layover time is worth an increase of $50 in plane ticket prices. Or I might state that travel time is more important than price, without mentioning specific values. Preferences can then be multivariable or single-variable. When I give a preference stating that the *ticket price*, the *service class*, and the *in-flight amenities* are more important than the *flight time* and the *distance to the airport*, I'm comparing three attributes to two. Finally, preferences can be over discrete or continuous attributes. Perhaps departing from one of a small set of airports is worth an increase of $50 in ticket prices. We will provide unified systems for representing and reasoning with each of these types of preference statements in this work.

It is beyond the scope of the current work to consider preference elicitation, which is the term given to the problem of coaxing users to reveal their preferences through introspection, interview, or observation. Preference elicitation is a hard and important problem. Although we do not treat this problem directly, it is because we are mindful of it that we approach our task with flexibility. By making no constraints on the amount or quality of preferences required to find a utility function, we give preference elicitors the maximum amount of leeway possible to approach their task.

1.1 Form of the Solution

Tradeoff preferences in our representation are interpreted as constraints on the partial derivatives of the utility function. For example, a decision maker could state that "Color is five times as important as price," a sentiment which one might express in the context of repainting one's home, and this is interpreted as indicating that utility increases in the positive color direction five times faster than utility increases in the positive price direction.

We show that using constraints on partial derivatives is a flexible and robust representation of tradeoff preferences. In particular, we show how to interpret each of the following:

1. Tradeoffs between concrete alternatives described partially, over many or few attributes,

2. Relative importance judgements of attributes, over many or few attributes,

3. Generalizations and exceptions to tradeoff preferences,
4. Qualitative *ceteris paribus*preferences between binary attributes,
5. Tradeoffs over continuous and discrete attributes,
6. Degenerate tradeoffs of a single attribute.

Further, we show how combine these statements to make a utility function that represents the given preferences, and can be used to quickly evaluate outcomes by attributes. For this utility function, we require however many or few preferences of whatever forms the decision maker can provide.

These tasks are the goal of this thesis.

1.2 Organization

In the following chapter, we first discuss some background and definitions that will be useful throughout. We define preference orders, utility functions, and give a more rigorous statement of what it means for a utility function to *represent* preferences. We also discuss market baskets and with operations upon them, their similarity to vectors. Chapter 2 ends with a presentation of our type of utility independence, generalized additive independence.

Chapters 3 and 4 are overviews of the related work; chapter 3 in the field of artificial intelligence and chapter 4 in decision analysis. Chapter 3 lays out our six criteria for a representation of preferences, and considers to what extent these criteria are fulfilled by existing preference reasoning and representation systems. We determine that there are large gaps in what existing systems can represent. Chapter 4 reviews techniques for attribute weight elicitation, the practice of assigning different numerical weights to each utility independent subset of the attributes.

Chapters 5, 6, and 7 introduce our language of preferences. Chapter 5 is devoted to the language of qualitative *ceteris paribus*preferences. Chapter 6 covers tradeoffs between partial descriptions of outcomes. Chapter 7 extends the treatments of tradeoffs between outcomes to tradeoffs between attributes. It is in these chapters that we present many of our results concerning the formal properties of our chosen representations.

Chapter 8 includes some techniques for determining exceptions and generalizations of preferences. We do not go into great detail on this subject in this thesis, rather we present one simple system and then outline how other theories of nonmonotonic reasoning might be applied to our preferences.

We present one method for constructing utility functions from the preferences presented earlier in chapter 9. This updates methods from earlier work in [MD04] to both support the enhanced representation of qualitative preferences in chapter 5 and reflect the addition of tradeoff preferences.

Chapter 10 shows how our tradeoff preferences of chapters 6 and 7 can be combined with a different type of qualitative *ceteris paribus* preference representation, the CP-net. The CP-net is one of the most popular choices for representing qualitative preferences. We show that straightforward techniques suffice to enhance a CP-net with quantitative tradeoffs.

Chapter 11 concludes with a review of our contributions and a discussion of promising avenues for future work.

Chapter 2

Background

In this chapter we present the basic concepts and notation that we use throughout.

We will first define a space of outcomes over which there exists a preference ordering. We then define a convenient shorthand for talking about partial descriptions of outcomes, the market basket. Then we can state the requirements of a preference order and of a utility function respecting that order. For describing statements about the preferences over market baskets, we then define a language of preference.

2.1 Preference Orders and Utility Functions

Let $A = \{A_i | 1 \leq i \leq n\}$ be a finite, enumerated set of attributes, and each attribute's domain be denoted D_i. Attribute domains can be infinite, discrete, or continuous. A set of outcomes is described by $\vec{A} = D_1 \times \ldots \times D_n$, a cartesian attribute space. When attributes have non-numeric domains, we assume that there is a function $\rho : \vec{A} \to \Re^n$, an isomorphism giving a numeric representation of the entire attribute space. Just to simplify the presentation and discussion in the remainder, we will assume throughout that the domain of each attribute is numeric. Although we make this simplification to the domains, whether the domains of each attribute are continuously differentiable or not will become of central importance.

A user or decision maker is assumed to have a preference ordering over the outcomes described by $\vec{A}$. The preference ordering is a reflexive and transitive relation $\succsim$ on $\vec{A}$ where $\vec{a} \succsim \vec{a}'$ indicates that $\vec{a}$ is at least as preferable as $\vec{a}'$. *Strict preference* $\succ$ consists of the irreflexive part of $\succsim$, that is $\vec{a} \succ \vec{a}'$ just in case $\vec{a} \succsim \vec{a}'$ but $\vec{a}' \not\succsim \vec{a}$. When $\vec{a} \succsim \vec{a}'$ and $\vec{a}' \succsim \vec{a}$ we say $\vec{a}$ is *indifferent* to $\vec{a}'$ and write $\vec{a} \sim \vec{a}'$. This is not assumed to be a total order. If two outcomes are incomparable we write $\vec{a} \bowtie \vec{a}'$.

A utility function, $u : \vec{A} \to \Re$, allows the use of $\geq$ as the standard order (and therefore preorder) over the reals, and thus over the image of $\vec{A}$ under u. We write $\geq_u$ for the preorder induced by u under $\geq$ over the set $u(\vec{A})$. Complete preorders $\succsim$ over countable $\vec{A}$ can be expressed exactly by utility functions, so that $u(\vec{a}) \geq u(\vec{a}')$ if and only if $\vec{a} \succsim \vec{a}'$. We say a utility function u *represents* a complete preference order $\succsim$ when $u(\vec{a}) \geq u(\vec{a}')$ if and only if $\vec{a} \succsim \vec{a}'$. An incomplete preorder $\succsim$ is necessarily a subset of some preorder $\geq_u$. When $\succsim$ is a subset of the preorder $\geq_u$, we say that u *is* *consistent with* $\succsim$. We denote by $\mathcal{U}(\vec{A})$ the set of possible utility functions $u : \vec{A} \rightarrow \Re$.

In the following chapters we will develop a language of preference statements called $\mathcal{L}(A)$, and extent it to languages $\mathcal{L}(A)_1$ and $\mathcal{L}(A)_2$. Each statement of preference in these languages will be interpreted as implying constraints on the utility function or on the possible preference orders consistent with the preference. Thus, for any statement $S \in \mathcal{L}(A)$, in $\mathcal{L}(A)_1$ or in $\mathcal{L}(A)_2$, and any utility function u in $\mathcal{U}(\vec{A})$, we can say that $u \models S$ if the constraints implied by S hold on u. In such cases we say that *u is consistent with S*. A *utility proposition* expressed by a sentence S is the set of utility functions consistent with S, written $[S] = \{u \in \mathcal{U}(\vec{A}) | u \models S\}$. If M is a set of statements S, we write $[M]$ for $\cap_{S \in M}[S]$. A utility function *u is consistent with M* iff $u \in [M]$.

We can state the definition of other common concepts relating languages and utility functions.

Definition 2.1 (Satisfiability) *A set of preference statements M is* satisfiable *iff $[M]$ is non-empty.*

Definition 2.2 (Conflicting Preference Sets) *Given two sets of tradeoff statements S and R, R* conflicts with *S and S* conflicts with *R if S is satisfiable and $S \cup R$ is unsatisfiable.*

Finding a utility function u consistent with a preorder $\succsim$ involves demonstrating two things. First, we must show what the function u computes: how it takes a model and produces a number. Second, we show and how to compute u given the preorder. We will use preferences, in various forms, as constraints on possible preorders.

2.2 Partial Outcomes

We will use *market baskets* as partial descriptions of outcomes. A market basket (or sometimes just "basket" for brevity) is a partial function from $\vec{A}$ to $\coprod_i D_i$, the disjoint union of domains of each attribute. For a market basket b, $b(i)$ is the value assigned by b to attribute i. $b(i)$ is either undefined, in this case we write $b(i) = \perp$, or $b(i)$ is a value w in D_i. If a basket defines one value for every attribute $A_i \in A$ then we say it is *complete*. We can also write a basket as a list of the pairs it defines: $b = \{(i = w_i), (j = w_j), \ldots\}$ where i indicates attribute A_i and $w_i \in D_i$. When a market basket b assigns values to every attribute in a set $G \subseteq A$ and for $i \notin G$, $b(i) = \perp$, we say that G is the *support* of b.

We define operations on market baskets by defining component-wise operations using standard algebra, but with additional support for $\perp$. When performing operations on $\perp$, we have, for any real r, $\perp + r = \perp$ and $\perp * r = \perp$. Otherwise, basket addition is much like vector addition, as is multiplication of a scalar and a basket. Multiplication of two baskets is component-wise: for baskets b, b', we have $b * b' = b''$ where $b''(i) = b(i) * b'(i)$. Replacement of a basket by values from another is written $b[b'] = b''$ and defined as follows: $b''(i) = b'(i)$ unless $b'(i) = \perp$, in which case

$b''(i) = b(i)$. We also write $b[(i = w)]$ for the replacement of b with an implicit market basket $b' = \{(i = w)\}$.

Market baskets and vectors are similar. For each market basket, we can define a corresponding vector, which we call the *value vector* for the basket. If b is a basket we write $v(b)$ for the value vector of b, where $v(b) \in \Re^n$. We use $v_i(b)$ for the i^{th} component of $v(b)$. Now, $v(b)$ is the value vector for b iff $v_i(b) = b(i)$ whenever $b(i) \neq \perp$ and $v_i(b) = 0$ otherwise. This correspondence allows us to use operations on vectors in place of operations on market baskets. For example, we will use the dot product of value vectors to compute the dot product of baskets: $v(b) \cdot v(b')$. On the other hand, we will also define the modification of a value vector to be the modification of the basket.

In the following we will also use componentwise vector comparisons, as follows. If there is a set of attributes $G \subseteq A$ with $m = |G|$, if $\vec{g}, \vec{g}' \in \vec{G}$, then $\vec{g} \geq \vec{g}'$ if and only if $g_i \geq g'_i$ for all $1 \leq i \leq m$. Similarly, $\vec{g} > \vec{g}'$ if and only if $\vec{g} \geq \vec{g}'$ and there is some $i, 1 \leq i \leq m$, where $g_i > g'_i$. Lastly, we will call a vector $\vec{x} \in \vec{A}$ the *characteristic vector for* G if $\vec{x}$ is such that $x_i = 1$ iff $A_i \in G$ and $x_i = 0$ otherwise.

2.3 Utility Independence

Utility independence (UI) is a property that obtains when the contribution to utility of some attributes can be determined without knowledge of the values assigned to other attributes. More precisely, if a set of features G is utility independent of a disjoint set of features G', then the utility given to G does not depend on the values assumed by the features of G'. We give some general background regarding utility independence presently.

We will need to talk about projections from the whole set of attributes to smaller sets, given a set of attributes $G \subset A$. The cartesian product of the domains of attributes in G is $\vec{G}$. We denote the projection by $\pi_G : \vec{A} \rightarrow \vec{G}$.

We will write vectors next to each other when we refer to their combination; if X and Y are disjoint sets of attributes, with $\vec{x} \in \vec{X}$ and $\vec{y} \in \vec{Y}$, then $\vec{x}\vec{y} \in \overrightarrow{X \cup Y}$.

Definition 2.3 (Utility Independence) *A set of features X is* utility independent *of a disjoint set of features Y with $A = X \cup Y$, if and only if, for all $\vec{x}, \vec{x}' \in \vec{X}$, and $\vec{y}, \vec{y}' \in \vec{Y}$,*

$$\vec{x}'\vec{y} \succ \vec{x}\vec{y} \implies \vec{x}'\vec{y}' \succ \vec{x}\vec{y}'. \tag{2.1}$$

We call this utility independence; Keeney and Raiffa [KR76] call it *preferential independence*, and use a different notation.

Note that utility dependence is not symmetric. It is possible for a set of features X to be utility dependent upon a disjoint set of features Y, while Y is independent of X.

Since utility independence generally simplifies the structure of the corresponding utility functions, in later chapters we assume that attributes are utility independent whenever there is no evidence to the contrary.

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a *cover* of the attributes A, that is, each $C_i \subseteq A$ and $\cup_i C_i = A$, and two attribute subsets C_i need not be disjoint. Given a cover $\mathcal{C}$, and some element of that cover C_i, we denote by $\vec{C}_i$ the cartesian product of the features in C_i.

We will call functions $\hat{u}_G : \vec{G} \rightarrow \Re$ partial utility functions; these take a partial description of an outcome and return a number. We define *subutility functions* to be functions $u_G(\vec{a}) = \hat{u}_G(\pi_G(\vec{a}))$, which ignores all but some set of features, G.

Given some cover $\mathcal{C}$ of A, a *generalized additive utility function* for $\mathcal{C}$ is a utility function $u : \vec{A} \rightarrow \Re$ that is a weighted sum of k subutility functions $u_i : \vec{A} \rightarrow \Re$ [BG95]. That is,

$$u(\vec{a}) = \sum_{i=1}^{k} t_i u_i(\vec{a}) \tag{2.2}$$

is a generalized additive utility function for cover $\mathcal{C} = \{C_1, \ldots, C_k\}$.

The generalized additive independent utility function plays a central role in this thesis. In the following we will try to find a generalized additive utility function for any input set of preferences; and we will do so for several different types of preference statements. These types of preferences will be explained and considered in the following chapters. Following those, we will discuss how to arrive at the utility function from the given preferences. But first, we will consider much of the other work related to our own, and try to demonstrate how our work extends that body.

Chapter 3

Preference Representation and Representational Criteria

Making decisions on behalf of human users or assisting humans with difficult decision-making tasks is a useful function computers might perform. Most existing approaches create preference reasoning formalisms the human must then use to express his or her preferences. Our objective is to lessen the burden on the human user by defining a variety of preference expressions that a user can use. Other proposed reasoning systems allow users to state preferences as utility functions, as *ceteris paribus* comparisons, and as weights on logical formulae. Our work will extend this by building on a base of more expressive *ceteris paribus* statements [WD91], and augmenting with techniques to reason with partial or incomplete information of both quantitative and qualitative preference information, including tradeoffs and preferences restricted to a given context. Our work also supersedes previous work on joint representations by providing a principled semantics for the various preference expressions in terms of utility functions. Importantly, this gives us a way to judge if our methods are correct, reasoning sound, and conclusions justified.

When we consider existing preference reasoning formalisms, we should consider the following issues. Existing systems are usually built to handle only one or two of the following.

1. Basic Qualitative Comparisons. A decision maker may wish to state simple comparisons. I may wish to make no explicit quantification of preference or utility, leaving the preference purely qualitative. I may wish to say, that other things being equal,

    - $pet(cat) \succ pet(dog)$

    which is to say, I prefer to have a cat than to have a dog as a pet. Such statements should be treated like constraints on admissible utility functions, requiring a utility function representing the decision maker's attitudes to value *pet(cat)* higher than *pet(dog)*. Other details of the interpretation are potentially variable from one system to another.

2. Complex Preferences. When stating preferences *ceteris paribus*, users may wish to make comparisons among more than one or two variables. For example, I may desire chocolate with butterscotch over strawberry. Further, quantitative variables complicate this scenario drastically: a user may have preferences between formulae, I may desire $p + \frac{1}{2}q$ over $3r$. Thus, preference reasoning systems should not needlessly restrict the syntax of a user's preference expressions to containing relations upon only one variable. Users must be able to express their preferences in as complex a manner as may be required.

3. Utility Independence. When a decision maker can identify that his or her preferences for values of attribute X do not depend on the values assumed by attribute Y, then preferences for X and Y can be assessed independently. This is known as the *Utility Independence* of X and Y. For example, the *price* of a car may be independent of the *color*: if we are choosing between two blue cars, we choose the cheaper. On the other hand, a canonical example of dependence (explored in [Bou94a]), is that the utility of carrying an umbrella depends on the current weather.

   Decision makers can have preferences that may or may not exhibit utility independence. Existing preference reasoning systems force a decision maker to declare which features are utility independent before stating any of their preferences. Under many situations, users may wish to make preference statements without such restrictions. I may wish to state that I prefer strawberry to chocolate, without any other stipulations, including whether or not this preference depends on my choice of icecream toppings. Therefore, any preference reasoning system containing syntax for expressing the utility independence or dependence of features on each other should perform equally well when such information is *not* provided.

4. Tradeoffs. Users may express quantitative tradeoffs between variables. I may prefer strawberry to chocolate by a factor of five. Or I may value finishing my thesis this year twice as much as finishing next year. Such a preference can be given a *ceteris paribus* interpretation, where holding other things equal the preference for thesis finishing dates holds. Thus a tradeoff statement should be able to express a desire indicating that an increase of some fixed amount of one feature (finishing date) offsets an increase of some fixed amount of another feature (thesis quality) by a factor of X. Someone might reasonably write that "Finishing Date > 5 times Thesis Quality."

5. Context. There are two potentially separate but related types of context-dependent preferences. First, preferences may be conditional, that they do not hold in general but only if some condition is true. If I say that my preference for fuel-efficiency over crash-test safety of vehicles holds only when the vehicle is not an SUV, I am conditioning my *fuel-efficiency* vs. *safety* tradeoff preference on the value of the SUV attribute. Second, preferences that hold in general my be overridden by preferences that apply in exceptional or emergency situations. I might make a general statement like

- $invest(stocks) \succ invest(bonds)$

indicating a preference for investing in stocks over investing in bonds. And then I might wish to state a special-case preference:

- $invest(bonds) \wedge recession(true) \succ invest(stocks) \wedge recession(true)$

indicating that if it is true that the economy is in recession, then I prefer investing in bonds rather than stocks.

6. Incomplete Preferences. Preferences over possible outcomes can be considered a pre-order, a transitive and reflexive relation, over the space of outcomes. But reasoning systems must be prepared to reason with only a partial specification of this pre-order. Users may be unable or unwilling to order every possible outcome. In general, the time required to do so is prohibitive. Thus preference reasoning systems must be able to make recommendations and decisions with as little or as much information as the user is able or finds it convenient to provide.

This list of preference criteria is not intended to be exhaustive. There may well be other authors who would choose somewhat different criteria. For example, other authors have considered the uncertainty of actions in their preference reasoning; indeed, this is a major dimension of utility theory and analysis in economics and operations research. However, in the context of preference reasoning and representation, there is much to be done before attempting to model the uncertainty associated with probabilistic actions, as evidenced by the criteria we present here. Thus while we consider the application of preferences to uncertain choices to be an important avenue of future research, we feel the field is better served by first solidifying the foundations of preference representation and reasoning in the case of certain actions. We maintain that all of those mentioned here are important capabilities for a preference system, and designing a system that fulfills them is a major step forward.

In the following we consider existing work on preference representations, and relate them to the preceding considerations.

3.1 Varieties of Preference Representation

Successful preference representations have been based on modeling utility functions and computing maximum expected utility, following economic theory of humans as rational decision makers. This approach is effective, but can be labor- and information-intensive. A decision maker and decision analyst would have to take time to assess utilities by means of standard gambles [vNM44] in a manner described by [KR76], and the necessary probabilities must be estimated. The effectiveness of this approach is intimately tied to the accuracy of the utility and probability estimates.

Systems have also been built using logical formalisms, where a user might specify logical formulas over variables describing outcomes and then indicate preferences by assigning weights, priorities, or orderings among such formulae [LvdTW02]. While such systems can be versatile, they can also present considerable cognitive overhead, making it difficult to tune the formulas precisely enough to provide the desired behavior.

Other systems use assumptions about the structure of a user's preferences to achieve computational efficiency gains. By forcing a user's preferences to obey additive utility independence, the resulting utility function can be efficiently computed. Some researchers assume additive independence (for example, [BG95]), some multiplicative independence [LS99]. Others assume that a user's preferences each concern exactly one feature of the domain [BBHP99]. Assumptions of such structure are not always well-founded, and although they can assure efficient computation with the preferences, they can prevent a user's true preferences from being properly expressed.

While many existing preference reasoning systems are adequate to the tasks they were designed for, they lack a significant focus on letting the user express preferences in a way that is most natural to the user. Psychological research shows that rewording the same underlying question can cause people to give opposite answers [TK86]. Thus, as designers of decision-support systems, we can expect that our expectations about the form or structure of a user's preferences can alter these preferences. In some cases, it may be desirable to coerce the user into expressing preferences in a way that makes them computationally efficient to reason about. But in many cases, this results in the user being unable to express their preferences as they might occur under undisturbed introspection. Further, users have many different kinds of preferences, and each system separately assumes that users preferences are of only one type (although [BD02] is a system assuming two types.)

Our work seeks to extend the capabilities of current decision systems by building a preference representation that is expressive and powerful enough to perform in complicated domains. Our work seeks to fit the representation to the user, not the other way around. In the following we explore representations of common preferences and their accompanying reasoning methods, and the computational properties of such methods.

3.2 Multi-Attribute Utility Theory

Economists, beginning with [vNM44], have studied decision making using utility functions. As we have mentioned, such representation are powerful but require much information or data to be effective. For example, [KR76] describes methods by which a decision specialist can phrase and pose hypothetical preference questions to a decision maker. This is known as *preference elicitation.*

To avoid time consuming preference elicitation procedures, [CKO01] exploits existing databases of utility functions (created over the history of decision analysis for certain medical domains) to gain prior distributions over a new user's utility function. With a few questions Chajewska, Koller, and Ormoneit can quite quickly determine which of a few common utility functions a user's preferences most closely mimic.

In many ways traditional numeric utility functions as developed in decades of research in economics are the gold standard of expressiveness for preference reasoning systems in AI. In fact, traditional economic utility functions succeed in many of the criteria we have presented for preference reasoning systems. Our reasoning system will try to come as close to the expressiveness of a utility function as possible while remaining simple to elicit from users and efficient to compute.

Utility models can allow us to make statements of the form $cat \succ dog$ but only by associating different numerical weights to different kinds of pets, such that the utility, u, of pets obeys $u(cat) > u(dog)$. Traditional economic utility functions do not allow qualitative preferences. Thus utility functions partially satisfy criterion 1, Qualitative Comparisons. Utility functions can be constructed that obey arbitrary preferences over complex alternatives or arbitrary dependencies or independencies. However, the effort required to do so can be prohibitive. Thus traditional utility functions pass criteria 2 and 3, Complex Preferences and Utility Dependence. The quantitative tradeoffs of criterion 4, Tradeoffs, can also easily be worked into utility functions

However, a utility function cannot allow more specific preferences (in the stocks and bonds example of criterion 5, Context) to override more general preferences. To model such a scenario, investing must be made utility dependent on the growth of the economy and one utility function must be constructed for the cartesian product *(investing, economy_growth)*. Utility functions are also notoriously poor with incomplete information, and have no real semantics for *ceteris paribus* preferences.

3.3 Logics of Preference

Logical statements of the form $p \succ q$, meaning formula p is preferred to formula q, can be given varying interpretations. Many such interpretations are qualitative. This allows decision makers more flexibility in expressing incomplete or fewer preferences or leaving strengths of preferences unspecified. This flexibility streamlines the preference elicitation process, allowing more natural and perhaps simpler preference statements to be made and made sense of.

3.3.1 Preferences Keeping Other Things Equal

One class of preference representations, termed *ceteris paribus*, allows preferences that apply "keeping other things equal." Such a preference might be "Other things being equal, I prefer chocolate to strawberry ice cream." These preferences capture the intuitive idea that other unmentioned qualities might affect the decision making process. If, for example, chocolate turns out to be much more expensive than strawberry, and hence all else is not equal, then this preference no longer applies.

Doyle, Shoham, and Wellman [DSW91] propose reasoning systems based on the *ceteris paribus* preference semantics. Their system states preferences as: $pet(cat) \succ pet(dog)$, meaning other things being equal, I prefer to have a cat than a dog. This explicitly compares the case of having a pet that is a cat (and not having a dog) to the case of having a pet that is a dog (and not having a cat), following [Han89]. Note that this says nothing about the preferability or desirability of cases where one has two pets, a dog and a cat, and states where one has no pets at all. These are left out of the preference semantics.

This system has several benefits. It firstly does not require or exclude the possibility of utility independence between attributes. Syntax for expressing utility independence can be added if needed; for example Bacchus and Grove [BG96] make explicit statements of utility independence. They use a notation listing all the features that are independent of a particular preference: $a \succ_{c,d,e} b$ indicates that the preference for a over b is independent of the values that variables c, d, and e assume, holding them equal (in the *ceteris paribus* sense), yet possibly dependent on values of other variables f, g, h, etc., in the domain. Secondly, in a preference $p \succ q$ both p and q can be formulae, possibly referring to many actual attributes, or even including numerical functions or combinations of attributes. Thirdly, such a representation makes no explicit assumptions about what is required for reasoning to take place, so as many or as few preferences can be stated as are available. So a simple *ceteris paribus* representation satisfies criteria for Qualitative Comparisons (1), Complex Preferences (2), Utility Independence (3), and Incomplete Preferences (6).

However, the chief drawback of the *ceteris paribus* assumption is that *ceteris paribus* preference semantics cannot represent specificity of preferences. We cannot have more specific preferences override less specific ones, as in the stocks and bonds example of preference Context (criterion 5). Although it seems natural to assert both of these preferences, first that stocks are better than bonds as a general strategy, and the second that recessions reverse this ranking as an important special case, they cannot both be expressed simultaneously and consistently. The "other things equal" clause of the first preference subsumes the condition on *recession* of the second preference.

As a solution, some researches have proposed combining logical representations of *ceteris paribus* preference with explicit quantification of the "strength" of the preference. [TP94] consider each preference to be quantified, $p \succ q$ by ϵ, such that the utility of p exceeds that of q by at least ϵ. This combines the issues of explicit quantification explored below (Section 3.5.1) with those of *ceteris paribus* preferences.

Finishing the list of preference-representation criteria is the ability to present tradeoffs. Strict logical *ceteris paribus* representations are deficient in this area.

3.3.2 Utility Independence Diagrams

Boutilier, Brafman, Geib, and Poole introduce CP-nets, or Conditional Preference nets (named for the structural similarity to Bayesian Networks) in [BBGP97]. This is a representation where *ceteris paribus* preferences are expressed over features using a graph. Nodes in the graph correspond to single features and the directed edges in the graph represent utility dependence between features. Using this system, one can determine the preference for a particular feature given the values of the feature's parents in the graph. Such a system allows one to represent preferences such as:

- If I have steak, I prefer red wine over white wine,
- If I have fish, I prefer white wine over red wine.

In the above, the *food* feature is considered a parent of the *wine* feature, since the preference over *wine* depends on the value of *food*. Such a representation allows explicit preferences to be represented along with explicit utility independence and dependence, and thus fulfills preference criterion 1 for expressing Qualitative Comparisons. This representation also fulfills the conditional clause of our preference Context criterion by making all preference statements conditional. However, the main weakness of this representation is its inability to represent the complex preferences described in criterion 2. Since each preference is restricted to referring to one node in the graph, and each node represents one feature, preferences cannot refer to more than one feature at a time. This prevents statements such as $p \succ q$, *e.g.*, "I prefer having a cat to having a dog," if *having a cat* is logically independent of *having a dog*. We could represent this preference if *cat* and *dog* are two values for the *pet* variable, but this includes the assumption that we may only have one pet, and having a dog and cat are mutually exclusive.

A CP-net cannot remain ambiguous about utility independence. Features either are or are not independent and this determines the structure of the graph in the CP-net. Thus this does not satisfy preference criterion 3, Utility Independence. No provisions are made for tradeoffs or partial information, failing criteria 4 (Tradeoffs) and 6 (Incomplete Preferences). The CP-net makes up for its lack of expressiveness with considerable gains in reasoning efficiency.

3.3.3 Tradeoffs with CP-nets

In [BD02] Brafman and Domshlak introduce CP-nets with tradeoffs, or TCP-nets. A TCP-net is an extension of CP-nets that allows trade-off ratios between features to be represented. This means that the relative utilities of, to continue the previous example, *having a cat* and *having a dog* can be expressed in the network. This alleviates some of the single-feature preference problem, because trade-off ratios are a type of preference between two different features. Thus, one can say

- I prefer having a cat to not having a cat,
- I prefer having a dog to not having a dog,
- It is more important to me to satisfy the *cat* preference than to satisfy the *dog* preference.

Brafman and Domshlak's definition of *more important*, written $X \rhd Y$, means that it is more important to satisfy the preference regarding variable X than to satisfy the preference regarding variable Y. This allows a TCP-net to represent preferences between outcomes that vary on (exactly) two variables. Thus, if $X \rhd Y$, then an outcome $x_1 y_2 z$ is preferred to an outcome $x_2 y_1 z$, where $x_1 \succ x_2$, and $y_1 \succ y_2$, and z is an assignment to all remaining variables outside of $\{X, Y\}$. This means that a preference where more than two variables change cannot be stated (such as $x_1 y_2 w_1 \succ x_2 y_1 w_2$), due to the binary nature of the *more important than* relation. Thus the TCP-net goes one step farther than a plain CP-net in representing complex preferences of the kind described in criterion 2, Complex Preferences, but does not approach full generality.

Further, $X \rhd Y$ only expresses lexicographic importance (any small improvement in X outweighs any large improvement in Y.) They say their results generalize to non-lexicographic. However, the only other option they mention is annotating the "$\rhd$" arrow with a large table listing the complete preference ordering of all members of the joint domain XY. This is unsatisfying, firstly because there should be something possible to say between these two extremes, and secondly because this is computationally equivalent to joining the features X and Y into one joint feature XY, which is ostensibly the problem TCP-nets were introduced to avoid.

Consider an example. Suppose we wish to represent a lexicographic order on three binary variables, A, B, C, (this is the same order we would use treating the assignments to ABC as binary representation of numbers: $111 \succ 110 \succ 101 \succ ... \succ 000$). We first create a CP-net with three nodes, A, B, C, and no links between them. For each node we add a conditional preference table (CPT) for that feature, where the CPT for A is $1 \succ 0$, for B we say $1 \succ 0$ and for C we also stipulate $1 \succ 0$. Then we add two importance links, $A \rhd B$, and $B \rhd C$. The *ceteris paribus* clause of our preference on A gives us the following four preferences over outcomes:

- $111 \succ 011$
- $110 \succ 010$
- $101 \succ 001$
- $100 \succ 000$

The *ceteris paribus* clause of the preferences over B gives:

- $111 \succ 101$
- $110 \succ 100$
- $011 \succ 001$
- $010 \succ 000$

And finally, the *ceteris paribus* clause of preference over C gives:

- $111 \succ 110$
- $101 \succ 100$
- $011 \succ 010$
- $001 \succ 000$

The importance link, $A \rhd B$, gives the following two preferences:

- $101 \succ 011$

- $100 \succ 010$

The other importance link, $B \triangleright C$, gives the following two preferences:

- $110 \succ 101$
- $010 \succ 001$

All of the above preferences are a subset of the transitive closure of a lexicographic ordering of the three variables, however, this leaves unordered two outcomes that should be ordered: 100 and 011. In a normal lexicographic order, $100 \succ 011$. However, this cannot be stated in a TCP-net, because the two outcomes differ in more than two variables. Note that in general it is possible to insert another outomce between these, and thus have a transitive order among such outcomes (*i.e.,* $100 \succ 010 \succ 011$), but in a lexicographic order, no other outcome can be inserted between these two outcomes without violating other constraints of the lexicographic order.

TCP-nets also allow conditional relative importance. This is the idea that the relative importance of X and Y changes with the value of some other variable Z. They call the possibly many features Z the *selector set of* X, Y. This is an extremely general concept, and as such has some adverse ramifications for the the efficiency results the authors make. For example, they show an acyclic TCP-net is always satisfiable, but checking consistency is exponential in the size of the selector sets, or linear in the description of the TCP-net.

It should be evident that the TCP-net does not fulfill the entire representation criterion for tradeoffs, sice it does not treat numeric tradeoffs. However, this is perhaps a result of the qualitative focus of the original CP-net presentation. Even if we grant that the results in [BD02] do cover tradeoffs in sufficient generality, the TCP-net still leaves several criteria unsatisfied; preference criteria for Utility Independence, Context, and Incomplete Preferences.

3.3.4 Further CP-net Extensions

In [BBB01], a quantitative extension to CP-nets is discussed, called a UCP-net. A UCP-net allows a combination of utility independence information with *ceteris paribus* semantics and explicit utility values assigned to each preference. Such an approach may be warranted (as [BBB01] observes) in cases where uncertainty is an issue, probabilities are known, and reasoning in terms of expected utility is required. However, once one adds numbers to a CP-net, one has to make sure none of the numbers are too big or too small, otherwise it will violate the *ceteris paribus* stipulation of the CP-net semantics. Thus, many UCP-nets that could potentially be specified by users of a decision system do not actually qualify as UCP-nets because the parameters involved have to be carefully balanced. [BBB01] provides methods for determining if a specified UCP-net actually is a UCP-net, involving calculations using the "minspan" and "maxspan" of a feature, which is defined as the greatest difference in utility observed between two different outcomes given some assignment to a variable of interest and varying assignments to the parent features. These tools in hand, they check if the minspan of X is greater than the sum of the maxspans of the children of X. If so, the network is a UCP-net. But it is not obvious what is or isn't a UCP-net without performing such calculations.

The UCP-net's extensions to a CP-net actually make it fulfill fewer of our preference criteria. The changes are intended to allow reasoning about expected utility with the efficiency of a CP-net, and such probabilistic reasoning is an important and difficult task, but one we consider outside the scope of this thesis's work. Therefore, the UCP-net actually satisfies none of our preference criteria.

Wilson [Wil04] uses a circumscription like semantics for augmenting conditional *ceteris paribus* preference statements in CP-nets. $u : x \succ x'[W]$ means that given value u for variable U, we prefer value x to x' for variable X, as long as things outside of W are held equal, but W we really don't care about. Thus the values of W are allowed to be arbitrarily different. This new language is more expressive than the normal CP-nets. A CP-net cannot represent a preference like $x \succ x'[Y, Z]$, because the CP-net is restricted to representing preferences between outcomes that differ only on one variable, rather than both Y and Z. The circumscribed preferences are not quite general, since they still allow preferences only between one variable at a time. Suppose a user wanted to express the preference $a_1b_2 \succ a_2b_1$ using an augmented CP-net. The user could specify that $a_1 \succ a_2[B]$, *i.e.*, a_1 is preferred to a_2 no matter what values B assumes, but then the user is committed to also specifying $a_1b_1 \succ a_2b_1$, $a_1b_2 \succ a_2b_2$, and $a_1b_1 \succ a_2b_2$, which may or may not be desirable.

Although this extension allows specifying preferences between outcomes that differ on more than one (for augmented CP-nets) or two (for augmented TCP-nets) variables, it does not allow preferences that differ in principled ways, such as preferences that are not preferentially independent or obey independence assumptions other than additive independence, and preferences with tradeoffs involving more than two variables. Thus this augmentation handles the Qualitative Comparison criterion, but not Complex Preferences, nor general Utility Independence. The purpose of this representation is surely to address approximate reasoning, stating preferences where all else is held *almost* equal, by allowing the specification of some irrelevancies, and does so to a significant extent, but remains firmly in the realm of propositional logic statements. If augmented TCP-nets are considered, these solve our criterion on Tradeoffs as well as normal TCP-nets. Like the other CP-net formalisms, this augmentation requires all preferences in a domain to be completely specified before reasoning proceeds.

3.4 Deontic Logics and Preference Logics

Researchers in philosophical logic have also investigated logics of preference. In [Han89] Hansson defines a preference $p \succ q$ as applying in the worlds most similar to the current world. This idea gets around the problem of saying a *ceteris paribus* preference applies in all possible worlds, as long as they differ only on p and q. This idea is later refined in [Han96] where Hansson considers a *ceteris paribus* preference to hold only in the current context, where the worlds under consideration are similar enough to be subsumed under the "other things equal" clause of the preference.

This addresses our Context preference criterion by attempting to restrict the context of preference statements, but does so by offloading this problem to the problem of defining the relevant worlds under consideration. In a similar manoeuver, Hansson considers circumstances where one might wish *ceteris paribus* to hold other things almost equal, and determines that different worlds under consideration are those where the differences are large enough to matter. Such solutions are philosophically meaningful, but from the perspective of an applied computer system, the problem of filtering outcomes, choices, or decisions to be either restricted to the right context or different enough to be significant needs to be solved.

Researchers in artificial intelligence have also considered pure logics of preference. Boutilier [Bou94b] considers *ceteris paribus* preferences that hold only in the *most normal* worlds, relative to a given normalcy ordering over states of the world.

Clearly, these approaches allow direct comparisons to be stated, and so satisfy our preference criteria for Qualitative Comparison and Complex Preferences. The later formulations ([Bou94b] and [Han96]) also allow more specific preferences to override more general ones, provided that the specific or exceptional circumstance is "different enough" from the current context. Computation is not an explicit concern in philosophical logic, which may explain why the idea of utility independence has not been explored. Tradeoffs between attributes are also unexplored. By not considering reasoning methods or algorithms, these systems sidestep the preference incompleteness criterion, but it is nevertheless evident that reasoning of some sort could proceed with as few or as many preferences as a user wished to state.

3.5 Logical Formulae with Qualitative Priority

Dubois, Le Berre, Prade, and Sabbadin [DLBPS98] suggest a representation of qualitative preference concurrent with a representation of qualitative uncertainty. A preference in their system is any formula in propositional logic, with an attached "priority" weight. This priority is one element of a finite set of priority ranks. The set of ranks is a totally ordered set. For example, a preference might be

- *Pet(dog)*, priority β.

In this example, we desire to have a pet dog, with priority β. The strength of the preference depends on the ordering of β within the set of priority ranks, and on the priority assigned to the other preferences we might have. Suppose we also have a preference for

- *Pet(cat)*, priority δ.

Then to determine which of these is preferred, we can simply see if $\beta > \delta$ or if $\delta > \beta$.

Because this system (and the systems discussed below) use first order logic as their main representation, this allows almost anything to be stated, if sometimes only convolutedly. However, reasoning in unabridged logic can be intractable, so although everything can be said, it is not necessarily natural or efficient. For example, no direct preferences are expressible. These logical representations contain only *implicit* preferences. The preference for p over q must be inferred from the utilities or priorities assigned to p and q: which is more preferred is a consequence of which formula is given a higher rating. Thus such systems fail our criteria for Qualitative Comparison and Complex Preferences. Numerical tradeoffs, utility independence, and context dependence cannot be stated at all.

3.5.1 Weighted Logical Statements

Similar to the work by [DFP97], work by van der Torre and Weydert [vdTW98] uses propositional logic to represent preferences, and these preferences are assigned "utility" parameters. However, van der Torre and Weydert use numbers for the utility parameter. This parameter can be positive or negative, allowing either positive benefits for achieving a goal or negative penalties for failing to achieve the goal. These goals are also required to be conditionals (although the condition can be empty), thus a goal is a conditional statement something like "if ϕ obtains, try to do ψ with utility x" and "if ϕ' obtains, and we cannot do ψ', subtract utility y."

When a situation in the world obtains, the positive and negative rewards of each satisfied goal are summed together. This allows situations where multiple goals are satisfied to be compared. Adding weights together creates preferences over outcomes or choices of actions. For example, suppose we have the following preferences:

- I desire to have a dog, with Utility = 4.
- I desire to not have a fence, with Utility = 2.
- If I have a dog, I desire to have a fence, with Utility = 3.

Thus the most preferred scenario is to have a dog with a fence, where the utility is $4 + 3 = 7$, whereas having a dog without a fence earns utility of $4 + 2 = 6$. Clearly the implications of weighted preference statements are dependent upon the weights provided. In the example above, if the weights are adjusted slightly, different conclusions are obtained [LvdTW02]. Thus this system suffers from some of the problems of traditional numerical utility function representations: that it is difficult to precisely specify all of the utilities involved so the desired conclusions result. Further, if the desired conclusions are already known, what benefit is the reasoning system? If the desired conclusions are not known to the decision maker, and therefore one cannot be sure of the utility values provided, is it fair to say that the decision formulation *represents* the preferences of the decision-maker?

To extend this reasoning system to make decisions concerning entire outcomes, rather than just preferences over formulae, the "lifting problem" must be solved. This is a way of providing semantics for desires over worlds. If, for example, if I prefer p to q, then are all worlds satisfying p preferred to all worlds satisfying q ? Or does it mean that there exists some p world better than all q worlds? Or are all p worlds better than some q worlds? Choosing which of these quantifiers are used in the semantics is what is called the "lifting" problem. In [vdTW01], they explore several different possibilities for lifting parameters. This lifting parameter is constrained to be one of *max* and *min* for both the more preferred and less preferred alternatives. Thus one might require that the *maximal* p-worlds are preferred to the *minimal* q-worlds. Note that other preference reasoning systems circumvent this problem. Expected utility is a lifting parameter for decisions made under uncertainty where exact probabilities and utilities are known. And the *ceteris paribus* assumption is one way of solving the lifting problem. Under a *ceteris paribus* assumption, a p-world is preferred to a q-world only when the two worlds are exactly the same, outside of p and q.

A representation based on numerical parameters on logical formula cannot easily express tradeoffs between attributes. The logical formula can refer to several attributes at once, but to express a function of them would require individually stating each value for the function. Direct comparisons are inexpressible. Special cases can override general preferences because a decision maker can assign one utility value to p and different, not necessarily subsuming, values to $p \wedge q$, $p \wedge r$, $p \vee r$, etc. [LvdTW02]. This also removes the ability to state any utility independence between attributes. Thus this is one of a very few reasoning systems that satisfies preference criterion 5, representing overriding and context-dependent preferences to a significant extent. However, as mentioned above, this fails at criteria 1, 2, 3 and 4.

3.6 Preferences in Support Vector Machines

In an effort to avoid many of the preference representation problems discussed so far, and in fact, to avoid representation as much as possible, Domshlak and Joachims [DJ05] present a method of learning a user's utility function from a few statements using a Support Vector Machine (SVM). In machine learning, SVMs are quite common, and quite successful. The main idea in SVM learning is that data is not easily separable, but if the data can be projected into a very high dimensional space, it can be separated with a straight line. Through a clever mathematical coincidence, it is possible to choose the high-dimensional space and the transform from the input space with great craft, such that the simple operations required in the high-dimensional space (just a dot-product) can be computed by carefully-chosen operations (called "kernels" in the SVM literature) in the input space. This allows complex data to be classified with a linear separator in a high-dimensional space, without ever having to represent anything in that space.

So [DJ05] proposes to translate preferences of the form $x \succ y$, where x, y are formula over a space of n binary dimensions, into a space of size 4^n, where there is one attribute for each variable and each interaction between variables in the input space. One major benefit of this translation is that it trivializes utility independence concerns. Every utility function has a linear function in this larger space. Another major benefit is a tolerance to some slight errors or inconsistencies in the input preferences. If there are a few preferences or data points out of line, these can be dealt with using the standard "soft margin" techniques of SVM algorithms. It is clear that this method easily fulfills our criteria for Qualitative Preferences, Utility Independence, and Incomplete Preferences.

While there are many benefits, there are also some shortcomings. This represen-

Criteria **System**	1	2	3	4	5	6
utility functions	$\frac{1}{2}$	y	y	y	n	n
cp-[DSW91]	y	y	y	n	n	y
CP-nets	y	n	$\frac{1}{2}$	n	$\frac{1}{2}$	n
TCP-nets	y	n	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	n
UCP-nets	n	n	$\frac{1}{2}$	n	n	n
Aug-CP-nets [Wil04]	y	n	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	n
Philosophy [Han96]	y	y	n	n	y	y
Qual. Priority [DLBPS98]	n	n	n	n	n	y
Quant. Priority [LvdTW02]	n	n	n	n	y	y
SVM [DJ05]	y	$\frac{1}{2}$	y	n	n	y
Tradeoffs + *cp* (this thesis)	y	y	y	y	y	y

Table 3.1: The various preference-reasoning systems reviewed herein, and the preference criteria they fulfill. The "$\frac{1}{2}$" symbol indicates partial fulfillment of the criterion. Criterion 1 is Qualitative Comparison, 2 is Complex Preference, 3 is Utility Independence, 4 is Tradeoffs, 5 Context, and 6 Incomplete Preference.

tation does not fulfill preference criteria for Tradeoffs or Context. They do mention that there are modifications, presumably forthcoming, that allow tradeoff preferences to be given as input to the SVM. Further, the main dimensional translation relies on enumerating the state space of each variable, and as such is unable to handle continuous variables or discrete variables of large (or infinite) cardinality. Thus this partially satisfies and fails our criterion for Complex Preferences.

3.7 Summary

The existing preference reasoning systems fulfill some of our criteria, but none fulfill all. For criteria 1 through 6, we summarize which of the representations satisfy them in Table 3.1. We also include the system proposed herein, in the final row, a system based on a novel representation for tradeoffs combined with a representation of qualitative *ceteris paribus* preferences. We outline our new system below.

3.8 A New Representation

As we have shown, there are statements of preference that cannot be formally stated in existing preference reasoning systems. Preferences regarding numerical tradeoffs cannot be combined with qualitative statements of direct preference, *ceteris paribus* preference, or context-dependent preference. It is our belief that such a representation would constitute a useful extension to the field. The remainder of this work is the exposition of such a representation.

We present in later chapters a representation of preference tradeoffs based on partial derivatives of the utility function. We argue that this has clear semantics and intuitive appeal. Roughly, we consider tradeoff statements to be conditions on the partial derivatives of the utility function. When someone says "I prefer having a cat to having a dog," we interpret that as meaning that the utility is increasing in the *cat*-direction faster than it is in the *dog*-direction. This, in turn, has a mathematical interpretation in terms of the partial derivatives of the utility function; we present the full details in chapter 6.

Further, our system of tradeoffs builds on the foundation of qualitative preference statements of [DSW91]. This combination allows users to specify numerical tradeoffs alongside *ceteris paribus* preferences. This augments the capabilities of the Doyle, Shoham, and Wellman system to fulfill our preference criterion 4 for Tradeoffs, by handling qualitative and quantitative tradeoffs. The proposal we present also fulfills the Context criterion by allowing both conditional statements, and discussing ways of reasoning about exceptions and generalizations. Our additions also allow very complex tradeoffs to be stated, as far as we know, this is the most general form of tradeoff reasoning proposed. In addition to performing in the absence of utility independence information, we present a number of theoretical results that address the relationship between stated tradeoff preferences and utility dependence. Lastly, our proposal makes use of only as much information as the user is willing or able to provide, fulfilling criterion 6.

Chapter 4

Attribute Weighting in Additive Utility Functions

By far the most common representation of preference used in practice is the utility function. The most common type of utility function is an additive one: a utility function that is a weighted sum of subutility functions. In some cases this is just the weighted sum of the attributes describing outcomes. We believe functions of this form to be both the simplest and most important representation of preference. Because of this priority, we consider throughout this work how all of our representations of preference can be translated into additive utility functions and how all of our methods behave when applied to the same.

In the following we consider the history and impact of various techniques for constructing and eliciting additive utility functions used in decision analysis, economics, psychology, and operations research. This illustrates what types of preference considerations are easy or hard for people to think about or suggest, what techniques are used to construct additive utility functions, and what sorts of situations and constraints occur in practical decision analysis.

When using additive utility functions, of the form

$$u(x) = \sum_{i}^{n} t_i u_i(x), \tag{4.1}$$

there is a need for specifying the tradeoff weights, t_i in equation 4.1, between subutility functions. When the subutility functions, u_i in equation 4.1, are simple or obvious attributes this can be called attribute weighting. The attribute weight for a single attribute is frequently termed the *marginal utility* for this attribute in economics. A related economic term is the *marginal rate of substitution of X for Y*, for two attributes X, Y, and is the ratio of the marginal utility of X over the marginal utility of Y. Just as this problem is known by many names, there are many techniques for addressing this problem. We outline these techniques, from different fields, in the present chapter.

4.1 Linear Utility Functions

Before we talk about different ways of determining appropriate tradeoff weights in linear utility functions, we can make some remarks on the general idea of using a linear utility function.

Simple additive utility functions have had great success in practical decision theory and operations research. Although many authors have presented this opinion, Edwards [Edw77] presents this argument in great strength. "Theory, simulation computations, and experience all suggest that weighted linear averages yield extremely close approximations to very much more complicated nonlinear and interactive 'true' utility functions, while remaining far easier to elicit and understand," [Edw77]. Further, Edwards and Barron [EB94] claim that all dimensions they've ever encountered applying decision analysis either slope monotonically up or down, have an internal midpoint (very rarely), or have no underlying physical dimension and are purely judgmental (such as how much you like the *look* of a car). In such cases linear or piecewise-linear subutility functions are perfectly warranted. Edwards and Barron [EB94] sum up their strategy of using a linear approximation everywhere by naming it, perhaps ironically, "heroic approximation."

Edwards and Barron [EB94] argue that sensitivity analysis can be used to determine when the linear approximation is a bad idea. The adequacy of the linear approximation can be tested by asking questions about a very small improvement, and whether that improvement is more valuable at the low end of the scale or the high end. In this way, a decision analyst can estimate the partial derivatives of this dimension, and in a non-uniform way, by estimating at more than one point in the space. The ratio of greatest improvement (of some small, fixed amount) to least improvement can be used to estimate the curvature of the function. Nonetheless, Edwards further claims that when the value of a dimension is monotonically increasing or decreasing, the correlation between total value computed using a linear approximation and a true nonlinear model is upwards of 0.99, although this may be due to the aggregation of other dimensions in the utility function downplaying any particular dimension's importance.

When one dimension has increasing utility up to a point and then decreasing utility, a simple linear model is no longer appropriate for that dimension. Edwards and Barron [EB94] term this preference reversal a violation of *conditional monotonicity*, and claim that these reversals are easy to detect. This agrees with Keeney [Kee92] (p. 167) who suggests that a preference reversal probably means the decision has its dimensions of analysis poorly defined, and that some aggregate dimension is actually more appropriate. This aggregation strategy is an approach taken in recent preference reasoning systems in artificial intelligence, such as [BBHP99], and automated in [MD04].

An obvious counterexample to the linear attribute function is the economic principle of diminishing returns. Edwards [Edw77] argues that the economic law of diminishing returns is only important when outcomes are separable along dimensions (as in a commodity bundle, or market basket, commonly used in economics). This is presumably less frequently the case in the applied decision analysis of Edwards' experience, where users make real choices rather than simulated decisions based on economic formulae.

4.2 Basic Elicitation Methods

There are several main methods of tradeoff weight elicitation. These are: the Analytic Hierarchy Process (AHP) [Saa80]; Swing Weighting [vWE86]; Pairwise Weighting [KR76]; and SMART [Edw77] and advancements SMARTS and SMARTER [EB94]. In the following we provide a description of the basic tradeoff-weight elicitation methods.

There is of course a trivial, naive method of tradeoff weight assignment. In this naive method the decision maker assigns numbers to each attribute. These are the tradeoff weights for the corresponding attributes. This approach suffers from what I call the "magic number" problem: that it is difficult for users to assign numbers to attributes and be simultaneously aware of all the consequences of them. Did the user realize by making one assignment that the combination of attributes X, Y, and Z outweigh W and V? Was this a conscious decision? The myriad consequences of any weighting given in this manner make it unlikely to produce the intended results.

4.2.1 Pairwise

A successful traditional method of selecting parameter weights is to assess them by standard gambles [KR76]. Once the subutility functions are assessed, or at least normalized, and the best and worst values are known, a decision analyst can present *gambles* to a decision maker. These gambles present a tradeoff as the following problem: For what value of p would you be indifferent between:

1. An outcome with the best value for attribute a_1 and worst value for attribute a_2
2. A lottery that gives a p chance at obtaining the best possible outcome over $\langle a_1, a_2 \rangle$, and a $1-p$ chance at obtaining the worst possible outcome over $\langle a_1, a_2 \rangle$

The decision maker then states the probability for winning the lottery for which he or she is indifferent between the first proposal and the lottery. For this probability, p, the expected utility of both options are the same. Thus, the weight assigned to attribute a_1 is p.

Much of the development of this technique in [KR76] is devoted to choosing the right questions to ask. For a utility function with n scaling parameters, the best questions are a series that result in n equations for the n unknown parameters. This allows a minimum number of gambles to be posed to the decision maker while still providing enough information to uniquely determine all the scaling parameters. Sometimes $n-1$ equations are used with the extra equation being the normalization condition: $\sum_i^n t_i = 1$.

4.2.2 SMART

SMART, standing for the Simple MultiAttribute Rating Technique, is a formal 10-step process, as follows [Edw77]. 1) Identify whose utility is to be maximized. 2) Identify the decision or issues. 3) Identify the possible outcomes or entities. 4) Identify dimensions of value evaluation for entities. 5) Rank the dimensions in order of importance. 6) Rate dimensions numerically, preserving rank order. *Start with 10 on least important*, and proceed upward (emphasis added). Allow participants to reorder and remove inconsistencies. 7) Normalize importance weights. 8) Measure each entity on each dimension. 9) Calculate utilities for each entity. 10) Decide which entity is best. Step 6) is the defining characteristic of the SMART method.

While simple, SMART makes up the core of many more sophisticated elicitation techniques. And further, in many practical decision analysis applications simplicity is considered a strength, and for this reason applied techniques will bear considerable similarity to SMART [MHS05].

4.2.3 Range Sensitivity

There is a behavioral weight elicitation bias that has to do with the range of values particular attributes can assume, known as the *range sensitivity bias* or *range sensitivity principle* [KR76]. When one attribute has a very small range of values, it can be over-weighted if the elicitation procedure suggests a weight independent of the range. For example, if a decision maker is asked to consider hypothetical new jobs, varying on the two dimensions of *Salary* and *Time Off*, it is difficult to assign a weight to either dimension without knowing the variation present in the available options. If *Salary* ranges from \$44000 to \$45000 while *Time Off* ranges from 10 days to 25 days, *Salary* might receive less weight than *Time Off*. Were the range of *Salary* \$35000 to \$55000 this attribute would be much more important. Weighting methods that normalize each attribute's range to be the interval [0,1] and then attempt to value the attribute in general terms are particularly vulnerable to this bias.

The range sensitivity bias has been the subject of a variety of behavioral experiments. Fischer [Fis95] conducts experiments investigating whether or not decision-makers will (correctly) display range sensitivity in their assignment of attribute tradeoff weights. He found that the more salient the task makes the cross-attribute considerations, the more subjects will exhibit range sensitivity in weight determination. Earlier investigation focused on whether or not the range of an attribute helped to define the goodness or badness of a particular value on that attribute. Beattie and Baron [BB91] found that only in these cases were the attribute weights range-dependent. The result of [BB91] could also be construed as demonstrating heightened range sensitivity when the attributes are made more "salient" to the decision maker by the context of the decision. However, the type of "salience" explored in [Fis95] is a much more concrete focus on the tradeoff between two attributes, forcing a decision maker to explicitly choose between a good value for one attribute and bad value for another, and a very bad value for the first attribute and good value for the second.

4.2.4 Swing Weighting

The Swing weighting method attempts to value attributes by raising them from their lowest possible state to their highest possible state, and placing a value on this change [vWE86]. The change this creates in the overall utility function is then the weight assigned to this variable. This method attempts to correct for the range sensitivity bias (particularly as it is exhibited in the SMART method) by making the differences between the attribute's lowest value and highest value more salient during the weighting procedure.

In the canonical Swing method, the decision maker starts by considering a hypothetical alternative that has the worst value on every attribute. Then this alternative has its best attribute raised to its best level and this change is given a score of 100. The decision maker is asked to consider which attribute he or she would most prefer to increase to its best value next, and what value to place on this improvement. This proceeds iteratively until all attributes are assigned some weight.

Since the most important attribute is given weight 100, other attributes are then given lower weights. This tends to result in users selecting numbers like 80, 70, and 50. Although the attribute weights are then normalized to sum to one, this is a potential source of bias.

This procedure is otherwise similar to the SMART method, which shares authors with the Swing method.

4.2.5 A Return to SMART: SMARTS and SMARTER

Edwards and Barron [EB94] present two updates of the SMART method for linear additive utility elicitation. The main problem with previous version is the attribute scale dependence: the weight applied to an attribute depends on the range of possible values, even when those values are normalized. Thus SMARTS is introduced (SMART + Swing), using the swing weighting elicitation method instead of just the SMART method.

SMARTER uses the centroid of the space of possible weight assignments to obtain quantitative weights from a qualitative ordering on weights [EB94]. This eases elicitation by not requiring decision makers to actually weight attributes at all, but merely to rank-order them in importance. Specifically, if weights are ranked 1 through k^{th} most important, then the i^{th} weight gets weight: $(1/k)\sum_{j=i}^{k}(1/j)$, so the first is 1 + 1/2 + 1/3, the second is 1/2 + 1/3, the third just 1/3. After these assignments are made, all the weights can be normalized.

4.2.6 Analytic Hierarchy Process (AHP)

Introduced by Saaty in 1980 [Saa80], the Analytic Hierarchy Process has two main components: one, using a matrix of pair-wise importance ratios, and two, structuring the decision space hierarchically. The result is an additive utility function over the decision space, and the main goal of the process is to elicit the attribute weights in this utility function.

First, the method takes a matrix of pairwise ratio judgements of relative importance of the attributes. Element a_{ij} of the matrix represents the factor (call it k) by which attribute i is more important than attribute j, and thus $a_{ij} = t_i/t_j$. These factors are graded on a 9-point linguistic scale, where 1 = i and j are equally important, 3 = i weakly more important than j, 5 = strongly, 7 = demonstrably or very strongly, and 9 = absolutely. The analyst first asks the decision maker to make such judgements for each pair of attributes in the upper triangle of the matrix, and then the method fills in the bottom half of the matrix with the reciprocals of the upper triangle. One may object that a user asked to make these types of interrelated judgements cannot be expected to be perfectly consistent, and this is true, but the AHP has a mechanism to correct this, shown below.

A derivation follows. Let A be an n by n matrix, and t_i be the weight assigned to attribute i in the additive utility function.

$$
\begin{aligned}
&a_{ij} = (t_i/t_j),\\
&a_{ij}(t_j/t_i) = 1,\\
&\textstyle\sum_j^n a_{ij}(t_j/t_i) = n,\\
&\textstyle\sum_j^n a_{ij}t_j = nt_i,\\
&\textstyle\sum_i^n \sum_j^n a_{ij}t_j = n\sum_i^n t_i
\end{aligned}
$$

and that, in matrix notation, is $A \cdot \vec{t} = n\vec{t}$, which is the formula for an eigenvector $\vec{t}$ and eigenvalue n of A. This equation has a subtle property. This equation holds only when the decision maker's inputs are all perfectly consistent; when it does not hold, we have

$$A \cdot \vec{t} = e\vec{t},$$

where e is the eigenvalue for $\vec{t}$. Thus, the farther e is from n the greater the inconsistency in the user's inputs. This is a useful property for an interactive weight elicitation procedure, since this may prompt the decision maker to adjust his or her weighting to be more consistent.

There have been many further advancements on the original method. In fact, there is an entire conference devoted to the Analytic Hierarchy Process.

For example, [PH01] find that AHP's canonical method, using verbal descriptions to assess weights (such as "is significantly more important than") and then mapping these to a 1-9 scale, leads to greater inconsistency than a method that uses a more behavioral-based mapping from verbal description to numbers. The statements, after all, are highly subjective, and a subjective assessment of their numerical value seems warranted. Another interesting incidental result in [P̈99] is that the labelled, or odd numbered, members of the scale are up to twice as likely to be chosen as the even, unlabelled scalars. Presumably respondents are somehow influenced by the suggestion that the labelled ratios are more usual or appropriate than the "intermediate" ratios.

4.2.7 General Remarks

An empirical study of these weighting methods[1] found that Swing, Direct, and Pairwise weighting methods elicited mostly the same weights, while AHP and SMART methods were slightly different from the rest [PH01]. The same study found that 30% of people actually change which attribute is judged most important when using different elicitation techniques. It is not clear if one method results in "better" attribute weights than another method.

An interesting finding of behavioral researchers showed that anyone performing a complete pairwise attribute weighting, making $n(n-1)/2$ judgements, of a series of between two and five attributes in a perfectly self-consistent manner gave all the attributes equal weights [PH01]. This finding leaves little hope of humans describing their desires with very much mathematical coherence. It is the growing realization that this task is generally difficult for human beings that led researchers away from methods that directly require the decision maker to assign numerical weights to attributes. We have already mentioned one modern method, SMARTER, which takes as input only the ordinal ranking of attributes (a task much more likely to be done consistently by human respondents) and then uses some mathematical assumptions about the structure of the desired utility function to arrive at numerical attribute weights. In the rest of this chapter we consider more tradeoff-weight elicitation methods that allow decision makers more flexibility in the input they provide to the method.

4.3 Interval Utility Methods

Salo and Hamalainen [SH92] present a method allowing tradeoff constraints to be bounded by linear inequalities, then solving for possible tradeoffs using linear programming. This method, called PAIRS, allows decision makers to specify tradeoffs less precisely, and therefore allow for greater chances of consistency. When first developed, this method was applied directly to attribute hierarchies or attribute trees, where there are categories of attributes arranged in a hierarchy, and the importances of the higher attributes are functions of their member attributes. Recently this method has also been applied to SMARTS weighting [MHS05]. The PAIRS method is straightforward in procedure. First the decision maker provides intervals for each attribute weight, *i.e.,* each weight t_i is constrained to be in $[a_i, b_i]$ with $0 \leq a_i \leq b_i \leq 1$. The aggregation of these constraints are a system of linear inequalities that can then be solved directly using linear programming.

Sugihara, Ishii, and Tanaka [SIT04] allow interval value judgements to be given in Analytic Hierarchy Process (AHP) elicitation. This is as follows. First they obtain pairwise importance judgements in intervals. This defines a matrix of weight-intervals. They then obtain constraints by requiring that weights must sum to 1, be non-zero, and some must be larger than others through extrinsic ordering constraints. This defines a system of linear inequalities which can be solved. They prove a result that

[1] but using SMART [Edw77] rather than SMARTER [EB94], presumably because SMARTS and SMARTER use swing weighting and are therefore redundant with swing weighting in this experiment.

if the intervals admit a crisp ("crisp" in contrast to "fuzzy" or "interval-valued") solution they can find one with linear programming. While this solution is an approximation to the real solution, it is easy to obtain through computation of a least upper bound and a greatest lower bound on each of the attribute weights.

4.4 Fuzzy Methods

There are also a host of fuzzy methods for multicriteria decision making. For a review of the categories of these methods, see Ribeiro [Rib96].

One example is the proposal of Angilella, Greco, Lamantia, and Matarazzo [AGLM04], which uses approximate nonlinear programming to find fuzzy-logic weighting functions of tradeoffs, using discrete Choquet fuzzy-set integrals. The inputs required are a total order over the outcome space, a partial order on the importance of attributes, and a total list of the pairs of attributes having positive or negative interactions. Due to the stringent input requirements, this sort of algorithm is probably of more interest to fuzzy-set theorists than decision analysts. For an introduction to fuzzy integral theory in decision analysis, see Grabisch [Gra96].

4.5 Decision Analysis Tools

There are innumerable decision support programs described in the operations research literature, each attempting to compute utility functions from some kind of user input and judgements on the relative weights of attributes. We present a chronological outline of this development below.

Jacquet-Lagreze and Siskos [JLS82] present the UTA method, a method which uses linear programming to estimate the parameters of the utility function. Here UTA stands for the French "UTilité Additive." There are two main steps: 1) assessment of an optimal utility function, and 2) sensitivity analysis using a special linear program.

In UTA, each attribute is partitioned into r equal parts. Outcomes falling into the same partition element are equally good, and indistinguishable by preference on this attribute. Marginal utility for an outcome can be computed by linear interpolation from the endpoints of its containing partition element. Constraints are arranged so that the lower bound of a partition element must be less than the upper bound of the same partition element. From these, together with some constraints from orderings over outcomes, a utility function is definable using linear programming.

Sage and White [SW84] describe the ARIADNE system, a decision making system of thoroughly behavioral motivation. Their system is designed with behavioral principles and biases in mind, and allows debiasing where possible. An important consequence of this is that the system is iterative; ARIADNE makes suggestions and then the user refines his or her preferences.

The system itself is a linear-programming system of intervals on the tradeoff weights. The decision maker is allowed to give intervals where he or she thinks the tradeoffs weights are (and ARIADNE is the first system to handle interval weights), then the LP gives solutions respecting those and the criteria that they must sum to one. Importantly, attributes are allowed to be arranged hierarchically. The arrangement is given by the decision maker. The probabilities can also be given in intervals, but not if the tradeoff weights are also given in intervals; both together result in a non-linear (quadratic) programming problem.

The inputs from the decision maker are fourfold: 1) scores on lowest-level attributes (attributes are allowed to be hierarchically arranged), 2) tradeoff weights, 3) probabilities, and 4) relative risk-aversion coefficients.

Weber [Web85] describes a method called HOPIE that proposes to fuse information of two types: one, holistic orderings amongst outcomes; and two, attribute or criteria tradeoffs. HOPIE takes incomplete information and determines a set of utility functions consistent with it. Since the result of this analysis is many possible utility functions, dominance and general decision optimality is defined by having all these functions agree on one outcome.

Like UTA [JLS82] before it, the HOPIE algorithm partitions each criterion into subranges, and the decision maker gives preferences to hypothetical alternatives in the cartesian product of these partitions. This preference is either a number in [0,1] or it can be an interval in the same range. Further, the decision maker must make some pairwise judgements between hypothetical canonical alternatives. These rankings and intervals then define a system of linear inequalities, which is solved with linear programming.

Moskowitz, Preckel, and Yang [MPY92] provide a system called Multiple-Criteria Robust Interactive Decision analysis (MCRID). Their goal is to move beyond a full precise specification of attribute weights, weights on consequences, value functions, and utility functions. The general insight is to collect vague information that the decision maker is sure of rather than precise information that might be wrong or labile. They also have no problem with basing the decision on subjective probabilities, a philosophical stance Raiffa [Rai68] terms "the Bayesian approach," and tacitly underlies many other systems.

Based on the RID approach, MCRID is an extension to deal with multiple criteria. RID proceeds in three general steps. One, ranges of probabilities for events or orderings amongst their probabilities are elicited or obtained; but not all, just those the decision-maker feels confident about. Second, RID assess certainty equivalents for various payoff distributions required by the algorithm. Third, RID elicits preferences between pairs of complete alternatives. This suffices to define a linear program that determines the decision-maker's remaining subjective probabilities. However, this requires the utility functions to be known (they assume either linear or exponential: $U(x) = ax + b$ or $U(x) = -e^{-cx}$), and the conditional payoffs of actions to be known. The system is interactive because it only asks as many questions as are required to find a solution.

MCRID is similar. The main idea is to elicit direct attribute weights, orderings, partial orderings, and interval values, combined with certain preferences over outcomes. Criteria are arranged into a value tree so that weights are combined at each level of the tree. MCRID then uses first- and second-order stochastic dominance to rule out some alternatives, which is justified when the attribute weights obey a normal distribution.

MCRID contrasts itself with ARIADNE [SW84] favorably by noting that ARIADNE is not interactive and does not filter out dominated alternatives as a performance heuristic.

An IBM research project, ABSolute [BLKL01], is designed for evaluating source offers for business applications. WORA is the weight-elicitation procedure part of it. WORA lets users order certain subsets of the outcomes. These are called "loops". For each loop, a linear program can be defined that constrains the weights based on the ordering of outcomes. For example, when $u(s_1) > u(s_2) > u(s_3)$ we get two constraints in the Linear Program, $u(s_1) > u(s_2)$ and $u(s_2) > u(s_3)$.

[BLKL01] shows some results about how this converges reasonably quickly (15 iterations for 10 variables) in simulations. These simulations suggest that the number of distinct binary rankings required is the same as the number of dimensions. The authors claim that the WORA technique is best used when there are large numbers of attributes.

Modern systems in decision analysis sometimes borrow work from machine learning techniques. These try to learn the parameters of the utility function from a few examples, rather than have the user input them. In one such approach, Bohanec and Zupan [BZ04] present a system based on using their function decomposition algorithm HINT [ZBDB99] for decision-support. In this case, the Slovenian Housing Authority is making loans to people for houses. The problem is to classify housing loan applications into three tiers of worthiness. The goal of HINT is to reverse-engineer the concept hierarchy actually used to make these decisions. They use human experts to pick each intermediate concept from a set of several that HINT identifies as being good, in fact, the exact human experts that generated the test cases, a practice they term "supervised function decomposition." In this way, the expert, or decision maker, helps the algorithm arrive at what he or she believes to be the correct utility function.

4.6 Relation to our work

The preceding sections have mentioned many problems and approaches to dealing with human decision makers and their preferences. We keep these in mind as we develop our own preference reasoning methods.

The justification of using linear functions applies to some extent; linear function can be great approximations to more complicated functions. The shortcomings we see with the arguments for linear approximations, that diminishing returns occurs with separable attributes and that the impact of one variable is outweighed by the others, are domain dependent. In our work we generally assume linearity whenever we can, more specifically, whenever that assumption is not known to be inaccurate. But at the same time, we are readily able to deal with small or moderate amounts of nonlinearity in user's stated preferences and revealed utility function.

There are many examples here of the difficulties encountered in tradeoff weight elicitation. It seems clear that the problems encountered tradeoff elicitation are frequently those of forcing all of the user's inputs to be linear and consistent. When users are prompted to list tradeoff parameters for every feature, or every pair of features, these complete judgements are not likely to fit the structural constraints desired by decision analysts: that the weights all sum to one, or that the relative importance of two attributes be calibrated to the relative importance of the other attributes. Thus the user's inputs together with these assumptions are inconsistent. While there are techniques that allow a decision analyst to discover, assess, and diagnose inconsistencies in many tradeoff judgements, simpler applications will not have the resources or ability to make such adjustments. By removing these assumptions and allowing users to state as few or many tradeoffs as they might feel confident doing we can have greater confidence in the information provided. This is the approach we take in our work.

The work considered in this chapter has shown that human preference elicitation is a difficult task. Humans might have malleable and uncertain preferences, and getting a decision maker to express these in a usable and consistent form can be problematic. Our work in this thesis does not directly concern or address preference elicitation. But because we are mindful of the difficultly of this problem, we design our preference representation with whatever flexibility and generality we can. Our preference statements will allow qualitative or quantitative preferences and comparisons, complicated comparisons between multiple attributes, conditional statements, and we have no requirements for utility independence and completeness of the elicited preferences. The generality of our statements and methods allow more flexibility in preference elicitation. We leave preference elicitation up to the particular preference applications.

Chapter 5

Preferences Among Conditions

In many domains, qualitative rankings of desirability provide a more natural and appropriate starting point than quantitative rankings [WD91]. As with qualitative representations of probabilistic information [Wel90], the primary qualitative relationships often times immediately suggest themselves. Primary qualitative relationships also remain unchanged despite fluctuating details of how one condition trades off against others, and can determine some decisions without detailed knowledge of such tradeoffs. For example, in the domain of configuration problems one seeks to assemble a complicated system according to constraints described in terms of qualitative attributes, such as the type of a computer motherboard determining the kind of processors possible. Some user of a configuration system might prefer fast processors over slow without any special concern for the exact speeds.

We will consider a language $\mathcal{L}(A)$, defined relative to some set of attributes A, used to express preferences over propositional combinations of constraints on attributes of A. We present part of this language here. We limit ourselves in this chapter to the qualitative, *ceteris paribus* preferences between basic propositions over attributes. In the following two chapters we augment this language with quantitative tradeoffs and importance judgements.

Wellman and Doyle [WD91] have observed that human preferences for many types of goals can be interpreted as qualitative representations of preferences. Doyle, Shoham, and Wellman [DSW91] present a theoretical formulation of human preferences of generalization in terms of *ceteris paribus*preferences, *i.e.,* all-else-equal preferences. *Ceteris paribus* relations express a preference over sets of possible worlds. We consider all possible worlds (or outcomes) to be describable by some (large) set A of finite, discrete features. Then each *ceteris paribus* preference statement specifies a preference over some features of outcomes while ignoring the remaining features. The specified features are instantiated to some value in their domains, while the ignored features are "fixed," or held constant. A *ceteris paribus* preference might be "we prefer programming tutors receiving an A in Software Engineering to tutors not receiving an A, other things being equal." In this example, we can imagine a universe of computer science tutors, each describable by some set of binary features A. Perhaps $A = \{$*Graduated, SoftwareEngineering_A, ComputerSystems_A, Cambridge_resident, Willing_to_work_on_Tuesdays,* $\ldots\}$. The preferences expressed above state that, for

Feature \ Tutor	*Alice*	*Bob*	*Carol*
Graduated	false	false	true
A in Software Engineering	true	false	false
A in Computer Systems	true	true	false
Cambridge resident	true	true	true
Will work Tuesdays	false	false	true
⋮	⋮	⋮	⋮

Table 5.1: Properties of possible computer science tutors

a particular computer science tutor, they are more desirable if they received an A in the Software Engineering course, all other features being equal. Specifically, this makes the statement that a tutor *Alice*, of the form shown in Table 5.1, is preferred to another tutor *Bob*, also in Table 5.1, assuming the elided features are identical, since they differ only on the feature we have expresses a preference over (grade in Software Engineering). The *ceteris paribus*preference makes no statement about the relationship between tutor *Alice* and tutor *Carol* because they differ with regard to other features.

With this sort of model in mind, we consider how to formalize these statements.

5.1 *Ceteris Paribus* Preference Statements

When talking about qualitative *ceteris paribus* preferences, we assume these are preferences over discrete attributes. For the present chapter we will assume that each attribute in A is of finite discrete domain. In following chapters we explore the differences discrete and continuous attributes present.

We first define *atomic value propositions* in $\mathcal{L}(A)$. These are propositions on one attribute α and one value of that attribute, w, with $w \in D_\alpha$. The following are the possible atomic value propositions.

- Atomic Value Proposition (AVP) : $\alpha = w \mid \alpha \neq w \mid \alpha > w \mid \alpha < w \mid \alpha \geq w \mid \alpha \leq w$

These atomic propositions are useful for defining values for attributes, or talking about subsets of the domain of a particular attribute. It is, however, sometimes difficult to mathematically represent the constraint $\alpha \neq v$ for continuous domains. In the current case, with all of A discrete variables of finite domain, we can use the equivalence $(\alpha \neq v_i) \iff (\alpha = v_1) \vee ... \vee (\alpha = v_{i-1}) \vee (\alpha = v_{i+1}) \vee ... \vee (\alpha = v_k)$, although this is sometimes unwieldy.

Compound value propositions are boolean combinations of atomic value propositions.

- Compound Value Proposition (CVP) : AVP $\mid \neg$ CVP $\mid$ (CVP $\wedge$ CVP) $\mid$ (CVP $\vee$ CVP) $\mid$ (CVP $\implies$ CVP)

Note that with this formulation a market basket is shorthand for a conjunction of atomic value clauses, those corresponding to the values of attributes stipulated by the basket.

Next we define *qualitative ceteris paribus preference clauses* over compound value clauses. *Ceteris paribus* preference clauses are composed of a preference relation symbol, $\succ_{CP}$ or $\succsim_{CP}$, and two compound value clauses.

- Qualitative *Ceteris Paribus*Preference Clause (Q) : CVP $\succ_{CP}$ CVP | CVP $\succsim_{CP}$ CVP

For example we could write $(\alpha_2 = 3) \succ_{CP} ((\alpha_1 = 1) \wedge (\alpha_3 = 0))$. When people make preference expressions of this sort, using the connectives "and" and "or," there is frequently some ambiguity. If I say, "I prefer apples to bananas and pears," is that equivalent to two separate statements: "I prefer apples to bananas" and "I prefer apples to pears," or is it equivalent to the statement "I prefer apples to the combination of bananas and pears"? (And in this case, due to the domain, there is even more chance for ambiguity: does this last statement refer to a combination and admixture of the two fruits, bananas and pears?) Further, statements of preference involving negation of values can seem artificial: "I prefer red cars to not-blue cars," is perhaps unambiguous but somewhat unnatural.

Finally we define a *conditional qualitative preference.* Conditional qualitative preferences are just the combination of a value clause with a preference clause.

- Conditional Qualitative Preference (CQ) : CVP $\implies$ Q

For example, we can make statements $((\alpha_1 > 3) \wedge (\alpha_1 < 5)) \implies ((\alpha_2 = 3) \succsim_{CP} (\alpha_2 = 1) \wedge (\alpha_1 = 0))$. The intended interpretation is just that the preference clause Q holds in the parts of the space satisfying constraint *CVP*.

Conditional statements do not hold uniformly over all of $\vec{A}$, but only over a subspace of $\vec{A}$. Specifically, a conditional statement S holds where its compound value clause is satisfied. We discuss the satisfaction of clauses in the following section.

5.2 Meanings

In the above, we have given grammars for various preference statements. It remains to give these statements meanings here.

Atomic value propositions (AVP) are satisfied by different assignments to the attribute in question. A value x for attribute α satisfies an atomic value proposition $(\alpha = w)$ iff $x = w$. Thus we have $x \models (\alpha = w)$ iff $x \in D_\alpha, x = w$. Similarly, the other atomic clauses are satisfied in the usual algebraic ways. We define $x \models (\alpha \neq w)$ iff $x \in D_\alpha, x \neq w$. If ">" is defined for D_α, define $x \models (\alpha > w)$ iff $x \in D_\alpha, x > w$. If $>$ is not defined on D_α then no value $x \in D_\alpha$ satisfies a clause $(\alpha > w)$. Similar constraints apply to the clauses concerning $<, \leq, \geq$. When p is an atomic attribute proposition over attribute α, a basket $b \models p$ iff $v_\alpha(b) \models p$. Note that $\perp \not\models p$ for all propositions p.

Compound value propositions (CVP) are satisfied by the usual rules of boolean logic. If the indicated logical relation on atomic propositions is satisfied by basket b, then the compound value proposition is satisfied. For example, let p, q be AVPs. Then basket $b \models (p \wedge q)$ iff $(b \models p) \wedge (b \models q)$. As usual, we have $b \models (p \vee q)$ iff $(b \models p) \vee (b \models q)$. $b \models \neg p$ iff $(b \not\models p)$. Finally $b \models (p \implies q)$ iff $(b \not\models p) \vee (b \models q)$. When H is a compound value proposition $[H] = \{\vec{x} \in \vec{A} | \vec{x} \models H\}$.

The *support* of a compound value proposition, p, is the minimal set of attributes determining the truth of p, denoted $s(p)$. Two baskets b and b' are *equivalent modulo* p if they are the same outside the support of p. Formally, $b \equiv b' \bmod p$ iff $s(a) = s(p)$ and $b[a] = b'[a]$.

To interpret qualitative *ceteris paribus* preference clauses, we must consider the satisfying assignments of the component compound value clauses and consider the implications each satisfying assignment has for the preorder over outcomes. First let us consider a qualitative *ceteris paribus* preference clause $Q = p \succ_{CP} q$, and two baskets b, b' over A, such that $b \models p$ and $b' \models q$. Q is interpreted as meaning that $b \succ b'$ iff $b[a] = b'[a]$ where $a = s(p) \cup s(q)$, for every such pair b, b'. Similarly, a qualitative preference clause $Q = p \succsim_{CP} q$ implies that $b \succsim b'$ iff $b[a] = b'[a]$, for every such pair b, b'. In these cases we write $(b, b') \models Q$.

Put another way, consider the satisfying assignments of a each compound value clause in a given qualitative *ceteris paribus* preference clauses. For a compound value proposition p, let $[p]$ be the set of baskets b over $s(p)$ such that $b \models p$. Note that the set $[p]$ is finite since any compound value clause has a finite number of satisfying assignments over finite domains. Given a qualitative preference clause $Q = p \succ_{CP} q$, we interpret this as meaning that all $[p]$ are preferred, *ceteris paribus*, to all $[q]$. Thus if $b \in [p]$ and $b' \in [q]$, a qualitative *ceteris paribus*preference clause $p \succ_{CP} q$ is equivalent to the set of clauses stipulating $b \succ b'$ for all pairs b, b'.

Conditional qualitative preferences are just like qualitative preference clauses, except that they only hold over particular regions of the space. For a conditional qualitative preference, $Q = c \implies p \succ_{CP} q$, when $b \models p$ and $b' \models q$, we have $b \succ b'$ iff $b \models c$, $b' \models c$, and $b[a] = b'[a]$ where $a = s(p) \cup s(q)$.

5.3 From Qualitative to Quantitative

In the preceding text, we have given the definitions and tools we need to make statements of qualitative desires, and understand their meaning relative to a utility function or a preference order. However, purely qualitative comparisons do not suffice in all cases, especially when the primary attributes of interest are quantitative measures, or when outcomes differ in respect to some easily measurable quantity. For example, in standard economic models the utility of money is proportional to the logarithm of the amount of money. Quantitative comparisons also are needed when the decision requires assessing probabilistic expectations, as in maximizing expected utility. Such expectations require cardinal as opposed to ordinal measures of desirability. In other cases, computational costs drive the demand for quantitative comparisons. For example, systems manipulating lists of alternatives sorted by desirability might be best organized by computing degrees of desirability at the start and then working with these easily sorted "keys" throughout the rest of the process, rather than repeatedly re-determining pairwise comparisons.

In the following chapters we consider statements of quantitative tradeoffs and what possible meanings those may have, and how they can be integrated with the qualitative statements described above.

Chapter 6

Tradeoffs Between Attributes

In qualitative preference representations the kind of statements that are allowed are typically $P \succ Q$ and $P \succsim Q$, where P, Q, are either formulae over attributes [DSW91] or values of finite attributes [BBHP99]. Using this form, if Mike is considering buying a new computer, he can state a preference between chip manufacturers: *Intel* $\succ$ *AMD*, other things being equal.

Using derivatives to represent tradeoffs between different options has its root in economics. In economics, tradeoffs between two commodities have been studied in terms of the marginal impact of each commodity on utility. In such formulations, the partial derivative of the utility function, u, with respect to an attribute f, is known as the *marginal utility of* f. The marginal utility of f_1 divided by the marginal utility of f_2 is invariant under linear transforms of u, and is variously called the *rate of commodity substitution of* f_2 *for* f_1 and the *marginal rate of substitution of* f_2 *for* f_1 [HQ80].

Thus, if we wanted to stipulate, for a particular utility function u, that the marginal rate of substitution of f_2 for f_1 was everywhere at least r, we could write that for all $\vec{a} \in \vec{A}$

$$\frac{\partial u}{\partial f_1}(\vec{a}) \Big/ \frac{\partial u}{\partial f_2}(\vec{a}) \geq r. \tag{6.1}$$

This handles the usual case of tradeoffs between two variables. But perhaps a more accurate preference is more complicated. Mike might state a preference between conjunctions saying that *AMD and Overclocked* $\succ$ *Intel and Overclocked*, again, other things being equal.

Quantifying these tradeoffs is a natural extension. If someone tells you that he or she prefers P to Q the next question you might ask is "By how much?" A reasonable and flexible expression of this is to quantify a tradeoff statement by some factor $r \geq 1$, and write statements $P \succ rQ$. Mike might say that *AMD* $\succsim$ 2 *Intel* if he thinks an AMD chip is twice as desirable as an Intel chip. This statement of preference can be viewed as a constraint on the possible utility functions consistent with the preference. This constraint can, in turn, be represented as an inequality among directional derivatives of a continuous utility function over continuous attributes (and we can generalize to discrete attributes later). A directional derivative of a function $u : \Re^n \rightarrow \Re$ evaluated at a point $\vec{a} \in \Re^n$ is the derivative along a vector with base

a and direction $\vec{x}$. Furthermore, the directional derivative of a function u in the direction of vector $\vec{x}$ is equal in value to the inner product of the gradient of the function with $\vec{x}$. This quantity measures the increase of u in the direction of $\vec{x}$. Thus if we want to talk about constraints on the directional derivatives of the utility function, or rates of increase in the directions of $\vec{x}$ and $\vec{y}$, we can state constraints of the form

$$\nabla u(\vec{a}) \cdot \vec{x} > r \nabla u(\vec{a}) \cdot \vec{y}, \tag{6.2}$$

for $r > 0$. All that remains to relate this to a preference expression like $P \succ rQ$ is to choose $\vec{x}$ to somehow represent P, and $\vec{y}$ to somehow represent Q. We explain how this is done in the following section.

Suppose we are modeling the following situation. Let there be two attributes, f_1, f_2, such that the preferences over f_2 depend on the values of f_1. In this scenario f_1 represents having fish or meat for dinner, and f_2 represents having red or white wine with the meal. A common stipulation is that if one is having fish, then one prefers white wine, but if one eats red meat, then one prefers a red wine accompaniment. Let u_1 be a subutility function representing the utility derived from the meal, including both f_1, f_2. Then suppose that f_3 represents the time it takes to arrive at the restaurant. Generally one prefers a shorter transit time, and there may well be tradeoffs people are willing to make between the quality of the restaurant and the time required to get there.

This situation sets up a tradeoff between one attribute and a group of attributes. Tradeoffs between two single attributes are straightforward: one compares a specified or normalized amount of one attribute to the other, which is exactly how the marginal rate of substitution is used. When considering tradeoffs between attribute sets, we consider these to be between the subspaces defined by the attribute sets: one compares a fixed or specific measure of increase in $\{f_1, f_2\}$-space to a corresponding measure in $\{f_3\}$-space, in this example. This will again be accomplished using inequality (6.2), and we will explain the necessary mathematical underpinning in the following.

Using conditions based on the differentiability of the utility function requires we consider what happens when the functions is not differentiable, and what happens when the domain of the function is discrete. We will show it is straightforward to use a discrete difference analog of the directional derivatives in the formulation of inequality (6.2).

Lastly, we consider a degenerate for of the tradeoff expression between attributes: the tradeoff between two different values of a single attribute.

In the following we will develop the ideas introduced here, using directional derivatives to express a variety of different preferences. We go on to present representations for a variety of tradeoff preferences between concrete alternatives, importance statements relating the relative value of attributes, and *ceteris paribus* (other things being equal) preferences. We show that these representations generalize both the economic notion of marginal rates of substitution and previous representations of preferences in AI.

6.1 Expressing Tradeoff Statements

We define a new language $\mathcal{L}(A)_1$ of tradeoffs over the attributes A. We present statements of tradeoff preferences between market baskets. We will use some of the same constructions from $\mathcal{L}(A)$, and repeat these two here:

- Atomic Value Proposition (AVP) : $\alpha = w \mid \alpha \neq w \mid \alpha > w \mid \alpha < w \mid \alpha \geq w \mid \alpha \leq w$
- Compound Value Proposition (CVP) : AVP | $\neg$ CVP | (CVP $\wedge$ CVP) | (CVP $\vee$ CVP) | (CVP $\implies$ CVP)

Although the above leads to a very general logical language of preference tradeoffs over values of attributes, we will generally concentrate in this thesis on restricted cases of the language. These will be special cases of the more general language definitions we present. The first such restriction we will consider is the market basket:

- Positive Value Proposition (PVP) : $(\alpha = w)$
- PVP List (PVPL) : PVPL , PVP | PVP
- Market Basket (B) : { PVPL }

While our analysis is restricted to market baskets, we present the more general statements for language completeness.

Next we define *preference clauses* over compound value clauses. These preference clauses are potentially quantitative, in contrast to the qualitative preference clauses we defined in chapter 5. Preference clauses are composed of a preference relation symbol, $\succ$ or $\succsim$, and two compound value clauses. In the following, r is a scalar, a number $r \in \Re, r \geq 1$.

- Preference Clause (C) : CVP $\succ$ CVP | CVP $\succsim$ CVP | CVP $\succsim r$ CVP | CVP $\succ r$ CVP

For example we could write $(\alpha_2 = 3) \succ ((\alpha_1 = 1) \wedge (\alpha_3 = 0))$, much as we did with qualitative clauses.

The main case we consider in the following is the tradeoff between sets of baskets, for $r \geq 1$.

- Disjunctive Normal Form (DNF) : B | DNF $\vee$ DNF
- Basket Tradeoff (BT) : DNF $\succ$ r DNF | DNF $\succsim$ r DNF

Finally we define a *conditional tradeoff statement.* Conditional tradeoff statements are just the combination of a value clause with a preference clause.

- Conditional Tradeoff Statement (TS) : CVP $\implies$ C

For example, we can make statements $(\alpha_1 > 3) \wedge (\alpha_1 < 5) \to (\alpha_2 = 3) \succsim 3(\alpha_2 = 1) \wedge (\alpha_1 = 0)$. The intended interpretation is just that the preference clause C holds in the parts of the space satisfying constraint C. A conditional basket tradeoff is then

- Conditional Basket Tradeoff (CBT) : CVP $\implies$ BT

One special case we will consider is the degenerate tradeoff: not between different values of different attributes, but between values of the same attribute.

- Single Attribute Tradeoff (SAT) : CVP $\implies$ PVP $\succ_1$ r PVP | CVP $\implies$ PVP $\succsim_1$ r PVP

We give meaning to basket tradeoffs in the following. We postulate that not all expressible statements in $\mathcal{L}(A)_1$ are inherently valuable as preference statements, but this is a task we leave for future work.

For a basket tradeoff statement $BT = d \succsim rd'$, d and d' are in disjunctive normal form. Each conjunctive clause is a market basket. Let X be the set of conjunctive clauses in d and let Y be the set of conjunctive clauses in d'; X and Y are then sets of baskets. Then $d \succsim rd'$ is meant to be interpreted as the conjunction of statements for each possible pair of value vectors $v(x), v(y)$, where $x \in X$ and $y \in Y$, "utility increases in the $v(x)$-direction at least r times faster than in the $v(y)$-direction." Thus basket tradeoff statements are interpreted as $|X| * |Y|$ separate statements about relative increases in different direction of u.

As before, conditional statements do not hold uniformly over all of $\vec{A}$, but only over a subspace of $\vec{A}$. Specifically, a conditional statement $H \implies BT$, for H a compound value proposition and BT a basket tradeoff, holds at $[H]$.

Statements S in $\mathcal{L}(A)_1$ make constraints on the shape of utility functions. A basket tradeoff statement $V \succsim rV'$ with $\vec{x}$ and $\vec{y}$ satisfying assignments for V, V', respectively, makes the constraint that utility increases in the $\vec{x}$-direction at least r-times faster than in the $\vec{y}$-direction. The formal mathematical definitions come in the following sections.

6.1.1 A Utility-Based Tradeoff Semantics?

Before we present our semantics of preference tradeoff statements, we should consider the most simple and obvious choice of tradeoff semantics. Why not proceed as we did in the previous chapter, and define a tradeoff preference by the stipulation that utilities of one class of outcomes should be greater than the utilities of another class of outcomes? We show here that this idea has some serious drawbacks.

Let us consider two candidate meanings for statements of tradeoffs involving groups of attributes. Given a basket tradeoff $x \succ ry$, for baskets x and y, consider these two intuitive meanings.

1. Utility increases in the $v(x)$-direction r times faster than in the $v(y)$-direction,
2. $v(x)$ is preferred to $v(y)$ by a factor of r.

This thesis takes option 1 as its main principle. In this option we make explicit reference to the shape of the utility function, but not particular values of the utility function. The second choice fits semantics based on utility comparisons of various outcomes.

However option 2 is not suitable, and it is in fact unworkable. In the second option we talk about the difference in preference; this must mean a difference in utility. Option 2 means that the utility of $v(x)$ is r times more than the utility of $v(y)$. Thus the most literal mathematical interpretation for 2 is this:

$$u(v(x)) > ru(v(y)). \tag{6.3}$$

But this is bad. The first issue is that it should allow basic utility independence between related attributes and the remaining attributes. Let x and y be partial descriptions of the outcome space (that is, market baskets specifying values for some attributes, and leaving other attributes' values unspecified). If basket x is over $s(x)$ and basket y over $s(y)$, let $G = s(x) \cup s(y)$ be the set of attributes mentioned by x or y. If G is utility independent of its complement, then let $u_G(v(x))$ be the subutility function for attributes in G. Then inequality (6.3) must hold when k is a constant representing the utility contributed by the attributes outside of G:

$$u_G(v(x)) + k > r(u_G(v(y)) + k). \tag{6.4}$$

However, there is no way for inequality (6.4) to hold independent of k, since it simplifies to:

$$u_{xy}(v(x)) > ru_{xy}(v(y)) + (r-1)k.$$

This inequality can only hold independent of k when $r = 1$. This means that inequality (6.3) is incompatible with the utility independence assumptions we made above. Thus, we have started from inequality 6.3, assumed some utility independence, and found that inequality 6.3 is unusable. If we want to speak about quantified tradeoffs with $r \neq 1$, and have our tradeoffs obey simple decision theoretic principles, we must consider the other possible semantics for tradeoffs. In the following we present semantics of tradeoffs based on the a multiattribute interpretation of partial derivatives of the utility function.

6.2 Marginal Propositional Preferences

The most straightforward type of tradeoff preference users are likely to state is a tradeoff between two particular values of two separate attributes. This is something of the form "I prefer to have a laptop with 2GB of RAM than to have one with a 3.6Mhz processor," when 2GB is a value of the *RAM* attribute and 3.6Mhz is a value of the *processor* attribute. For generality, we allow the strength of the preference to be quantified by some factor $r \geq 1$, where someone might say "It is r times better to have a laptop with 2GB of RAM than to have one with a 3.6Ghz processor." Such statements can easily be given an interpretation in terms of the directional derivatives.

The above statement is, of course, a special case of the kinds of statements allowed in our language of tradeoff preferences, $\mathcal{L}(A)_1$. In general, users may wish to state tradeoffs involving more than two attributes at a time. Suppose we wish to state that a 3.6Ghz processor with a 1-MB L2-cache and a 800Mhz front side bus is twice as good as 4GB of ram with a 266Mhz front side bus. In this case, we are talking about values for four different variables, *processor, cache, bus,* and *RAM.* This also fits into our language simply; we can define market baskets x, y such that $x = \{(processor = 3.6\text{Ghz}), (cache = 1\text{MB}), (bus = 800\text{Mhz})\}$ and $y = \{(RAM = 4\text{GB}), (bus = 266\text{Mhz})\}$. Then to express this preference we can just say that $x \succ 2y$.

When we come to the interpretation of this statement as constraints on the utility function, we must be slightly more careful. We will try to state that "Utility increases in the $v(x)$-direction r times faster than it does in the $v(y)$-direction." However, we do not want the result to depend on which vector in the $v(x)$-direction we choose, therefore we need to take unit vectors in the $v(x)$-direction.

The following general definition captures this intuition for tradeoff statements.

Definition 6.1 (Conditional Basket Tradeoff) *Let S be a conditional basket tradeoff $H \implies d \succsim rd'$ in $\mathcal{L}(A)_1$, where d, d' are in disjunctive normal form. The meaning of S is the following constraints on the utility function. Let X and Y be the set of conjunctive clauses in d and d' respectively, and then let $x \in X$ and $y \in Y$ be market baskets, then $u \in [S]$ iff for all pairs x, y,*

$$\left(\nabla u(\vec{a}) \cdot \frac{v(x)}{|v(x)|}\right) \bigg/ \left(\nabla u(\vec{a}) \cdot \frac{v(y)}{|v(y)|}\right) \geq r. \tag{6.5}$$

holds on all $\vec{a} \in [H]$.

Basket tradeoff statements make conditions on the partial derivatives of the utility function. In particular, another way of writing the condition in inequality (6.5) is as follows. For all points $\vec{a} \in [H]$, if $u \in [x \succsim ry]$ then

$$\sum_{x(i) \neq \perp} \frac{\partial u}{\partial f_i}(\vec{a}) \frac{v_i(x)}{|v(x)|} \geq r \sum_{y(j) \neq \perp} \frac{\partial u}{\partial f_j}(\vec{a}) \frac{v_j(y)}{|v(y)|}. \tag{6.6}$$

We continue the example from the beginning of this section, regarding preferences over computers. First let us assume that the *processor* dimension is measured in gigahertz, the *cache* and *RAM* dimensions measured in megabytes, and the *bus* measured in megahertz. Then inequality (6.5) makes the following constraint on the utility function

$$\frac{\partial u}{\partial processor}(\vec{a}) \frac{3.6}{\sqrt{640013.96}} + \frac{\partial u}{\partial cache}(\vec{a}) \frac{1}{\sqrt{640013.96}} + 800 \frac{\partial u}{\partial bus}(\vec{a}) \frac{800}{\sqrt{640013.96}} \geq$$
$$2\left(\frac{\partial u}{\partial RAM}(\vec{a}) \frac{4}{\sqrt{70772}} + \frac{\partial u}{\partial bus}(\vec{a}) \frac{266}{\sqrt{70772}}\right).$$

Some remark is warranted about the large denominators in the formula above. Each variable has different ranges of values, and if values of different variables are compared, then the ranges play a role. The above inequality is a constraint on the partial derivatives of the utility function. This constrains attributes with smaller values to have more weight, which compensates for attributes with larger values. We know from chapter 4 that normalizing attribute ranges is not always an appropriate action, since this leads to bias and misunderstanding. Thus leaving the constraints to balance the attributes is preferable to normalizing the attributes beforehand.

Conditional basket tradeoffs are also transitive, in the expected way. We state this in the following theorem.

Theorem 6.1 (Transitivity) *Two conditional basket tradeoffs*

$$S_1 = H_1 \implies \vec{x} \succ r_1\vec{y},$$
$$S_2 = H_2 \implies \vec{y} \succ r_2\vec{z}$$

together imply a third tradeoff statement

$$H_1 \wedge H_2 \implies \vec{x} \succ r_1 r_2 \vec{z}. \tag{6.7}$$

Proof. Given two statements S_1, S_2 as above, these are interpreted as the following constraints

$$S_1 : \nabla u(\vec{a}) \cdot (\vec{x}/|\vec{x}|) > r_1 \nabla u(\vec{a}) \cdot (\vec{y}/|\vec{y}|),$$
$$S_2 : \nabla u(\vec{a}) \cdot (\vec{y}/|\vec{y}|) > r_2 \nabla u(\vec{a}) \cdot (\vec{z}/|\vec{z}|)$$

If we multiply S_2 by r_1 we can then substitute back into S_1, this gives the statement in equation (6.7). We arrive at the condition $H_1 \wedge H_2$ by observing that the conjunction of statements S_1, S_2, holds only in the intersection of their conditions. This proves the theorem. □

We should note that the definition of quantitative tradeoffs between groups of features is a generalization of the standard economic notion of the marginal rate of substitution between two attributes. More precisely, the marginal rate of substitution is usually defined as the negative of the slope of the indifference curve of the utility function for two commodities [HQ80]. Specifically, in the case of unit-vector tradeoffs between two-attributes, our directional derivative representation for tradeoffs between sets of variables reduces to a condition on the the marginal rate of substitution. We show this by simplifying the condition in definition 6.1.

Theorem 6.2 (Reduction to Marginal Rate of Substitution) *When baskets x, y are $x = \{(i,1)\}$ and $y = \{(j,1)\}$, if u is consistent $H \implies x \succ ry$ then $\frac{\partial u}{\partial i} / \frac{\partial u}{\partial j} \geq r$.*

Proof. The definition 6.1 implies $\nabla u(\vec{a}) \cdot \frac{v(x)}{|v(x)|} / \nabla u(\vec{a}) \cdot \frac{v(y)}{|v(y)|} \geq r$. Expanding the dot product gives:

$$\frac{\partial u}{\partial i} \Big/ \frac{\partial u}{\partial j} \geq r.$$

And this is the constraint on the marginal rate of commodity substitution, as required. □

The description of tradeoff preferences we have given so far is very general. In research on preference elicitation, linear functions or simple piece-wise linear functions are considered exceedingly common [Kee92, EB94]. In these cases the partial derivatives of those utility functions have simple forms. We consider this in the following theorem.

Theorem 6.3 *Given a cover $\mathcal{C} = \{C_1, C_2, ..., C_Q\}$ of A, a generalized additive utility function $u(\vec{a}) = \sum_i^Q t_i u_i(\vec{a})$ for that cover, if u is linear in each $f \in A$ then for baskets x, y*

$$u \models (H \implies x \succsim ry) \Leftrightarrow \sum_{j=1}^{n} \sum_{i|j \in C_i}^{Q} t_i \frac{v_j(x)}{|v(x)|} \geq r \sum_{j=1}^{n} \sum_{i|j \in C_i}^{Q} t_i \frac{v_j(y)}{|v(y)|}. \tag{6.8}$$

Proof. By definition 6.1, a statement $H \implies x \succsim ry$, with baskets x, y, is equivalent to

$$\nabla u(\vec{a}) \cdot \frac{v(x)}{|v(x)|} \geq r \nabla u(\vec{a}) \cdot \frac{v(y)}{|v(y)|}.$$

If we expand the dot-product in the above expressions, we have

$$\sum_{j=1}^{n} \frac{\partial u}{\partial f_j}(\vec{a}) \frac{v_j(x)}{|v(x)|} \geq r \sum_{j=1}^{n} \frac{\partial u}{\partial f_j}(\vec{a}) \frac{v_j(y)}{|v(y)|}.$$

When $u(\vec{a}) = \sum_i^Q t_i u_i(\vec{a})$, the above is

$$\sum_{j=1}^{n} \sum_{i|j \in C_i}^{Q} t_i \frac{v_j(x)}{|v(x)|} \geq r \sum_{j=1}^{n} \sum_{i|j \in C_i}^{Q} t_i \frac{v_j(y)}{|v(y)|}.$$

This establishes the equivalence. $\square$

The summations in expression (6.8) simply count the number of times an attribute mentioned in baskets x, y occurs in one of the sets of attributes $C_i \in C$. Each time this occurs, there is an interaction between a value of interest and the subutility function for set C_i, so the tradeoff parameter t_i for that function must be included. This theorem will be of more importance later when we construct linear utility function from preference statements in $\mathcal{L}(A)_1$.

Note that theorem 6.3 means that a basket tradeoff statement applied to a linear utility function results in linear constraints on the parameters of that utility function. While it may seem odd to state constraints in this way, it is common for the form of the utility function to be known (and known or assumed to be linear) while the parameters of that function are yet to be determined. This determination proceeds according to the tradeoff preferences given by the decision maker.

6.2.1 Discrete Attributes

There is a natural extension of our directional derivatives formulation of tradeoffs to discontinuous utility functions over discrete attributes. When we have two market baskets of values x, y, and want to say that utility is increasing in the $v(x)$-direction r times faster than in the $v(y)$-direction, and the variables are discrete, we can still give meaning to this type of preference by using discrete difference analogues of the partial derivatives.

For market baskets x, y we define a *discrete difference vector* by

$$\Delta u(x,y) \equiv \begin{array}{c} \langle (u(v(x)) - u(v(x[(1 = v_1(y))])))/(v_1(x) - v_1(y)), \ldots, \\ (u(v(x)) - u(v(x[(i = v_i(y))])))/(v_i(x) - v_i(y)), \ldots, \\ (u(v(x)) - u(v(x[(n = v_n(y))])))/(v_n(x) - v_n(y))\rangle . \end{array} \tag{6.9}$$

Using discrete difference vectors, we can present a definition of discrete tradeoffs.

Definition 6.2 (Discrete Basket Tradeoffs) *Suppose S is a basket tradeoff $H \implies d \succsim rd'$, with d, d' in conjunctive normal form. Then let X, Y be the sets of conjunctive clauses in d and d' respectively. A utility function u is in $[S]$ if and only if for all pairs x, y, with $x \in X, y \in Y$, we have for all complete market baskets a, a', with $v(a), v(a') \in [H]$,*

$$\Delta u(a,a') \cdot \frac{v(x)}{|v(x)|} \geq r\Delta u(a,a') \cdot \frac{v(y)}{|v(y)|}. \tag{6.10}$$

The definition we have given for discrete tradeoffs is in an important sense equivalent to the definition we have given for continuous tradeoffs in the preceding section. We show that when u is a continuous linear function over continuous attributes, condition (6.10) implies consistency with the obvious basket tradeoff statement, $x \succsim ry$.

Theorem 6.4 (Linear Tradeoffs) *Suppose S is a conditional basket tradeoff statement $H \implies d \succsim rd'$, with d, d' in conjunctive normal form. Then let X, Y be the sets of conjunctive clauses in d and d' respectively. When u is a continuous linear function over continuous attributes, if u satisfies inequality (6.10) for all pairs x, y, with $x \in X, y \in Y$ and all complete baskets a, a', with $v(a), v(a') \in [H]$, then $u \in [S]$.*

Proof. By assumption we have

$$\Delta u(a,a') \cdot \frac{v(x)}{|v(x)|} \geq r\Delta u(a,a') \cdot \frac{v(y)}{|v(y)|} \tag{6.11}$$

for all pairs x, y in X, Y respectively and for all complete baskets a, a' in $[H]$. For linear u, the terms in the expansion of the discrete difference are merely the slopes in each direction of the utility function. Thus we have

$$u(v(x)) - u(v(x[(i = v_i(y))]))/(v_i(x) - v_i(y)) = t_i$$

for some value t_i representing the slope of the utility function in the i-direction. We can then write inequality (6.11) as

$$\sum_i^n t_i \frac{v_i(x)}{|v(x)|} \geq r \sum_j^n t_j \frac{v_j(y)}{|v(y)|}.$$

Then using linearity of u again, we can also substitute the partial derivatives of u (evaluated at any $\vec{a} \in [H]$) for slope parameters t_i and get

$$\sum_i^n \frac{\partial u}{\partial i}(\vec{a}) \frac{v_i(x)}{|v(x)|} \geq r \sum_j^n \frac{\partial u}{\partial j}(\vec{a}) \frac{v_j(y)}{|v(y)|}.$$

And this is the definition of $u \in [S']$. □

6.2.2 Single Attribute Tradeoffs

The langauge $\mathcal{L}(A)_1$ can be used to express tradeoffs within one attribute. Although not really a tradeoff between different variables, this can be thought of as a tradeoff between different values of the same variable, a degenerate case of the other tradeoffs mentioned so far.

Consider a single attribute tradeoff statement $H \implies x \succsim_1 ry$, where x, y are positive value propositions, or equivalently, baskets of a single attribute. Let this single attribute be Z. Then we define a single attribute tradeoff as follows.

Definition 6.3 (Single Attribute Tradeoff) *Given a single attribute tradeoff statement $S = H \implies x \succsim_1 ry$, where x, y are positive value propositions over attribute Z, a utility function $u \in [S]$ iff*

$$\nabla u(\vec{a}) \cdot v(x) \geq r \nabla u(\vec{a}) \cdot v(y), \tag{6.12}$$

for all $\vec{a} \in [H]$.

We have stated the condition of this definition in terms closest to definition 6.1 to highlight the main difference; in this case we deal with unnormalized vectors, while in definition 6.1 we normalize vectors. In the case of single attribute tradeoffs, both related vectors point in the same direction, so normalization would result in a relation between two identical vectors. In the general tradeoff case, the direction of the vectors is important.

Let us look in more detail at condition (6.12). Assume that baskets x, y are $x = \{(Z = z_1)\}$ and $y = \{(Z = z_2)\}$. Then inequality (6.12) becomes

$$\frac{\partial u}{\partial Z}(\vec{a}) z_1 \geq r \frac{\partial u}{\partial Z}(\vec{a}) z_2$$

which implies that either $z_1/z_2 \geq r$ or $\frac{\partial u}{\partial Z} = 0$. The second option means that utility does not depend on attribute Z, which would be unlikely given preferences concerning attribute Z. We are left, then, with the first option, a constraint on the values within the Z-attribute. While this is not a tradeoff between attributes, and is therefor of tangential interest in this thesis, we do give one illustration of how to deal with this type of constraints on the subutility function. Suppose that there are many such statements, and that they give us constraints like

$$\begin{aligned} z_1/z_2 &\geq r_1, \\ z_2/z_3 &\geq r_2, \\ z_3/z_4 &\geq r_3. \end{aligned} \tag{6.13}$$

Suppose now, without loss of generality, that $z_4 = 1, z_3 = 2, z_2 = 3$, and $z_1 = 4$. We can accommodate these constraints by using an exponential subutility function for attribute Z. Let $R = \max\{r_1, r_2, r_3\}$. Then let $u_Z(z) = R^z/R^4$. We will show that this subutility function satisfies the required conditions.

First let us call two single attribute tradeoff preferences $S_1 = \{(Z = z_1)\} \succ_1 r_1\{Z = z_2\}$ and $S_2 = \{(Z = z_2)\} \succ_1 r_1\{Z = z_3\}$, *linked basket tradeoffs* when they share a basket on opposite sides of a preference operator ($\succ_1$ or $\succsim_1$). There can be arbitrarily many linked basket tradeoffs; we call it a *chain of linked basket tradeoffs* when successive pairs of tradeoff statements are linked: if S_1, S_2 are linked, and S_2 and S_3 are linked, we call these S_1, S_2, S_3 a *chain.* We say it is an *increasing chain* when statements S_1, S_2, S_3, etc, are such that $z_1 < z_2 < z_3 < \ldots$.

Theorem 6.5 *Suppose we have a length-k increasing chain M of linked single attribute tradeoffs of the form $S_i = (Z = z_{i+1}) \succsim_1 r_i(Z = z_i)$ for i ranging from $1, ..., k$. Let $R = \max_i r_i$. If subutility function $u_Z(z)$ is of the form $u_Z(z) = R^z/R^k$ then $u \in [M]$.*

Proof. First consider the constraint on u implied by tradeoff statement S_i:

$$\nabla u(\vec{a}) \cdot v(\{Z = z_{i+1}\}) \geq r \nabla u(\vec{a}) \cdot v(\{Z = z_i\}).$$

This in turn is

$$z_{i+1} \frac{\partial u_Z}{\partial Z}(z_{i+1}) \geq r z_i \frac{\partial u_Z}{\partial Z}(z_i)$$

and when we take the partial derivatives of u_Z we get

$$z_{i+1} \ln(R) R^{z_{i+1}} \geq r z_i \ln(R) R^{z_i}.$$

Simplifying gives

$$z_{i+1} R^{z_{i+1} - z_i} \geq r z_i,$$

next, let us assume, without loss of generality, that $z_{i+1} - z_i \geq 1$. We have $R \geq r$ by definition, so the above is then equivalent to

$$z_{i+1} R \geq r z_i,$$

which is true since $z_{i+1}/z_i > 1$ and $R \geq r$. □

It is straightforward to convert the above theorem and construction for a subutility function into a construction for a wrapper for an existing subutility function. We do not dwell on this further. It is also simple to extend this theorem for dealing with chains of linked basket tradeoff preferences to other types of degenerate preferences; other types of preferences can be considered degenerate chains or sets of smaller chains. The theorem we present here covers the main case.

We make one more remark on the topic of degenerate tradeoffs. This remark is about what is *not* a degenerate tradeoff. When we have statements concerning multiple attributes, even though those attributes are the same, we have no real difficulties, this is just the same as any basket tradeoff already discussed. Consider a preference

$$S = H \implies x \succsim ry$$

where baskets x, y are over the same attributes. Assuming u is linear in $s(x)$ and in $s(y)$, then we have this condition:

$$t_x x_1 + t_y y_1 \geq r(t_x x_2 + t_y y_2)$$

If we assume that $x_1 \geq x_2$ and $y_2 \geq y_1$ then we have

$$t_x(x_1 - r x_2) \geq t_y(r y_2 - y_1)$$

which is just a linear inequality in two unknowns, t_x, t_y, just as any other tradeoff preference we have dealt with before.

Chapter 7

Importance of Attributes

We have so far considered tradeoffs between particular instances of variables: an assignment to some variables X is preferred to an assignment to some variables Y. In this chapter we define tradeoffs between groups of attributes, which can also be considered a statement about the *importance* of the groups of attributes.

When we talk about the importance of the variables involved, we are talking about the relative influence in our utility function. This importance is a quantity that does not depend on the direction involved, it does not depend on it being a positive or negative contribution to utility. Nor does the idea of importance we use depend on the particular values of attributes; it is purely a measure of the weight assigned to an attribute itself. This is, perhaps, primarily a difference related to the elicitation of preferences; sometimes users will want to talk about the relative value of attributes and sets of attributes.

The "importance" of attributes is a quantity that is frequently used in traditional decision analysis, in fact all of Chapter 4 is a review of various techniques for obtaining, assuming, or computing the relative importance of attributes in a utility function. Our presentation here extends the usual decision analysis methodology in three ways. The first is that the user is not required to make any importance statements at all. We merely take as many importance statements as occur and consider them alongside the other preference information we have. The second is that we do not expect users to talk about importance in just one attribute. The importance statements we describe herein are between sets of attributes. A user may decide that the combination of attributes *meal-quality*, *drink-quality*, and *atmosphere-quality* are at least twice as important as *time-to-arrive* and *time-spent-waiting* at a restaurant. And the third difference is that we consider importance statements between groups of attributes to be based on a comparison between the best outcomes in those attributes. We operationalize this by considering the relative importance between sets G and G' to be between the norms of the gradients of G and of G'.

7.1 Expressing Importance

To discuss our relative importance methodology in detail we augment the language of preference to handle statements of importance.

We now present a language $\mathcal{L}(A)_2$ for making tradeoffs between attributes themselves. In these cases we do not refer to specific values of attributes, just to the names of attributes. Thus, let us say that the descriptors $\alpha_1, \alpha_2, ..., \alpha_n$, represent the names of the attributes A. We then define an *atomic attribute term* as follows

- Atomic Attribute Term (AAT) : $\alpha_1 \mid \alpha_2 \mid ... \mid \alpha_n$

Further, we use $\perp$ for the empty set of attributes and $\top$ for the set of all attributes. An *attribute set* is defined as follows:

- Attribute List (AL) : ATT | ATT, AL
- Attribute Set (AS) : { AL } | $\top$ | $\perp$

Attribute clauses are then combinations of the usual set operations:

- Attribute Clause (AC) : AS | (AC $\cup$ AC) | (AC $\cap$ AC) | $\overline{AC}$

An *attribute tradeoff statement* is the combination of two attribute clauses and a tradeoff parameter $r \geq 1$ as follows:

- Attribute Tradeoff Statement (ATS) : AC $\rhd$ AC | AC $\trianglerighteq$ AC | AC $\rhd$ r AC | AC $\trianglerighteq$ r AC

Finally, a *conditional attribute tradeoff statement* is the combination of a compound value proposition with an attribute tradeoff statement using the implication operator $\implies$ as follows,

- Conditional Attribute Tradeoff Statement (CAT) : CVP $\implies$ ATS

Our main object of concentration in this chapter is conditional attribute tradeoff statements (CAT). The attribute clauses in attribute tradeoff statements have meaning by simplification. An attribute clause is just set operators applied to sets of attributes, and by the usual set operation definitions, each attribute clause is equivalent to some set G. When an attribute statement $C \rhd rC'$ is given, and attribute clause C is equivalent to set G, and similarly C' is equivalent to G', then the meaning of the statement is that "attributes G are more important than attributes G' by a factor of r."

We consider two special cases in this chapter. First we consider tradeoffs with discrete quantification. In the following $\alpha, \beta \in \Re$.

- Discrete Attribute Tradeoff (DAT) : AC $\rhd^{\alpha}_{\beta}$ AC | AC $\trianglerighteq^{\alpha}_{\beta}$ AC | AC $\rhd^{\alpha}_{\beta}$ r AC | AC $\trianglerighteq^{\alpha}_{\beta}$ r AC
- Conditional Discrete Attribute Tradeoff (CDAT) : CVP $\implies$ DAT

Discrete attribute tradeoffs are designed to capture the following linguistic assertion: an increase of at least α in the utility of attributes G is r times as important as an increase of at most β in the utility of attributes G'.

Second we consider tradeoffs of binary attribute comparisons, where we have

- Binary Attribute Comparison (BAC) : ATT $\rhd$ r ATT | ATT $\unrhd$ r ATT

which is just a relation between two attributes.

7.2 Marginal Attribute Importance

Tradeoffs between the attributes themselves can be represented in a manner quite similar to tradeoffs between values of attributes. The following definitions parallel our development in the previous chapter of marginal propositional preferences.

In the following definition we use the *absolute value of the gradient*, that is, the absolute value of each element in a vector, and we denote this by the function symbol $abs()$, where $abs(\vec{x}) = \langle |x_1|, |x_2|, ..., |x_n| \rangle$. We use this notation to distinguish the absolute value from the length of the a vector, $|\vec{x}|$. For the following statement, recall that a vector $\vec{x} \in \vec{A}$ is the characteristic vector for G if $\vec{x}$ is such that $x_i = 1$ iff $A_i \in G$ and $x_i = 0$ otherwise.

Definition 7.1 (Generalized Importance Statement) *Given a conditional attribute tradeoff statement $S = H \implies C \unrhd rC'$, with sets of attributes $G, G' \subseteq A$ equivalent to clauses C, C', respectively, let $\vec{x}$ and $\vec{y}$ be characteristic vectors for G, G', then a utility function $u \in [S]$ iff*

$$(abs(\nabla u(\vec{a})) \cdot v(x))/(abs(\nabla u(\vec{a})) \cdot v(y)) \geq r \tag{7.1}$$

holds on all points $\vec{a} \in [H]$.

Another way of writing the condition in inequality (7.1) is as follows. If $u \in [S]$ then for all points $\vec{a} \in [H]$

$$\sum_{f \in G} \left| \frac{\partial u}{\partial f}(\vec{a}) \right| \geq r \sum_{f \in G'} \left| \frac{\partial u}{\partial f}(\vec{a}) \right|. \tag{7.2}$$

This form will be of more use when we construct utility functions later.

Corollary 7.2.1 *Let S be a conditional attribute tradeoff statement $H \implies C \unrhd rC'$. Let sets of attributes $G, G' \subseteq A$ be equivalent to clauses C, C', respectively. S is satisfiable if $G \backslash G' \neq \emptyset$.*

Proof. In this case we have

$$\sum_{f \in G} \left| \frac{\partial u}{\partial f}(\vec{a}) \right| \geq r \sum_{f \in G'} \left| \frac{\partial u}{\partial f}(\vec{a}) \right|.$$

When $u(x) = \sum_{i=1} t_i u_i(x)$ the above is equivalent to

$$\sum_{f_i \in G} |t_i| \geq r \sum_{f_j \in G'} |t_j|.$$

We can exhibit a utility function $u \in [S]$ by choosing u as follows. Let f be an attribute in $G \backslash G'$. Then let t be the weight associated with attribute f. Then if we choose u such that $t > r \sum_{f_j \in G'} |t_j|$, $u \in [S]$.

Corollary 7.2.2 *Let S be a conditional attribute tradeoff statement $H \implies C \trianglerighteq rC$. Let set of attributes $G \subseteq A$ be equivalent to clause C. S is satisfiable only if $r \leq 1$.*

Proof. We have $H \implies G \trianglerighteq rG'$ which is

$$(abs(\nabla u(\vec{a})) \cdot \vec{x})/(abs(\nabla u(\vec{a})) \cdot \vec{x}) \geq r,$$

and this simplifies to $1 \geq r$. □

There is another geometric interpretation for attribute statements $H \implies G \trianglerighteq rG'$. The following is perhaps more intuitive and, we will show, equivalent to the definition already given. We prefer definition 7.1 for its similarity to definition 6.1, and present the following to motivate definition 7.1.

Given an arbitrary subset $G \subset A$ and a function over that $\vec{G}$, u_G, the gradient of u_G, $\nabla u_G(\vec{x})$, at some point $\vec{x} \in \vec{G}$ is a vector based at x pointing in the direction of maximum increase of u_G in $\vec{G}$. The length $|\nabla u_G(\vec{x})|$ of that vector is the magnitude of that increase. Thus if we interpret a tradeoff between a set of attributes G and another set of attributes G' as a comparison between the maximum possible rates of increase in the two vector spaces defined by G and G', we could write that in terms of the magnitudes of gradients in those spaces. Specifically,

$$|\nabla u_G(\vec{a})| / |\nabla u_{G'}(\vec{a}))| \geq r \tag{7.3}$$

compares the increase in the G-space to the increase in the G'-space. Further, if we choose the L_1 norm to measure the length of the above vectors, inequality (7.3) is equivalent to

$$\sum_{f \in G} \left| \frac{\partial u}{\partial f}(\vec{a}) \right| \geq r \sum_{f \in G'} \left| \frac{\partial u}{\partial f}(\vec{a}) \right|.$$

Let $\vec{x}$ and $\vec{y}$ be the characteristic vectors for G and G', respectively, then the above is equivalent to

$$\sum_{i}^{|A|} \left| \frac{\partial u}{\partial f_i}(\vec{a}) x_i \right| \geq r \sum_{i}^{|A|} \left| \frac{\partial u}{\partial f_i}(\vec{a}) y_i \right|,$$

and this in turn is equivalent to inequality (7.2). This derivation shows that the above measure of importance between attribute sets reduces to the directional-derivative representation of importance comparisons. This correspondence allows us to use the intuitive characterization of importance tradeoffs as comparisons of the maximum increase in two different spaces while extending the framework of partial derivatives presented in chapter 6 for the formal semantics.

Our definition of an attribute tradeoff ratio leaves open the possibility that the two sets of features involved are not disjoint.

Theorem 7.1 *For G, G' sets of features, $J = G \cap G'$, then $H \implies G \trianglerighteq rG'$ implies*

$$\sum_{f\in(G\backslash J)} \left|\frac{\partial u}{\partial f}(\vec{a})\right| \geq r \sum_{f\in(G'\backslash J)} \left|\frac{\partial u}{\partial f}(\vec{a})\right| + (r-1)\sum_{f\in J}\left|\frac{\partial u}{\partial f}(\vec{a})\right|$$

for all $\vec{a} \in [H]$.

We omit the proof, but it follows directly from definition 7.1. An important corollary of theorem 7.1 is the case of $J = G'$.

Corollary 7.2.3 *For G, G' sets of features, if $G' \subset G$ and $r \geq 2$, then*

$$H \implies G \trianglerighteq rG' \Leftrightarrow H \implies G \backslash G' \trianglerighteq (r-1)G'.$$

Proof. The proof is straightforward. By definition $G \trianglerighteq rG'$ is

$$\sum_{f\in G} \left|\frac{\partial u}{\partial f}(\vec{a})\right| \geq r \sum_{f\in G'} \left|\frac{\partial u}{\partial f}(\vec{a})\right|.$$

Since $G' \subset G$, we can split the first summation, giving

$$\sum_{f\in(G\backslash G')} \left|\frac{\partial u}{\partial f}(\vec{a})\right| + \sum_{f\in G'} \left|\frac{\partial u}{\partial f}(\vec{a})\right| \geq r \sum_{f\in G'} \left|\frac{\partial u}{\partial f}(\vec{a})\right|.$$

From this it is obvious that we have

$$\sum_{f\in(G\backslash G')} \left|\frac{\partial u}{\partial f}(\vec{a})\right| \geq (r-1) \sum_{f\in Y} \left|\frac{\partial u}{\partial f}(\vec{a})\right|,$$

which establishes our equivalence. □

Just as with basket tradeoffs, attribute tradeoff statements reduce to linear conditions on the parameters of linear utility functions. We state here a similar theorem to theorem 6.3, but for attribute tradeoffs. Recall that for a basket b, $s(b)$ is the support of b, or the set of attributes assigned values, other than $\perp$, in b.

Theorem 7.2 *Given a cover $C = \{C_1, C_2, ..., C_Q\}$ of A, a generalized additive utility function $u(\vec{a}) = \sum_{i=1}^{Q} t_i u_i(\vec{a})$ for that cover, and arbitrary baskets x, y, if u is linear in each $f \in A$ then*

$$s(x) \trianglerighteq rs(y) \Leftrightarrow \sum_i \sum_{f\in s(x)\cap C_i} t_i \geq r \sum_j \sum_{f\in s(y)\cap C_j} t_j. \tag{7.4}$$

The proof of this theorem is omitted; it follows the proof of theorem 6.3. Again, the idea of this theorem is to count the number of times an attribute mentioned in attribute sets $s(x), s(y)$ occurs in one of the sets of attributes $C_i \in C$.

Just as tradeoff statements are transitive, so are statements of conditional importance.

Theorem 7.3 (Attribute Transitivity) *Two conditional attribute tradeoff statements*

$$\begin{aligned} H_1 &\implies G \rhd r_1 G', \\ H_2 &\implies G' \rhd r_2 G'' \end{aligned}$$

together entail the tradeoff statement

$$H_1 \wedge H_2 \implies G \succ r_1 r_2 G''.$$

Proof. Omitted, but parallels theorem 6.1 exactly.

7.2.1 Importance of Discrete Attributes

We can again extend preferences concerning attribute importance over continuous utility functions to preferences over discontinuous utility functions, as we have done with preferences concerning particular value tradeoffs. In this case, we give semantics to the discrete attribute tradeoff statements, which are intended to model this intuition: an increase of at least α in the utility of attributes G is r times as important as an increase of at most β in the utility of attributes G'. We show that when this concept is applied to continuous attributes and continuous linear utility functions, this coincides with the earlier definition of attribute tradeoffs in definition 7.1.

We formalize importance of discrete attributes in the following definition, capturing the intuitive meaning just mentioned.

Definition 7.2 (Discrete Importance Statements) *For any $G, G' \subseteq A$, and any $\alpha > 0$, $\beta > 0$, a utility function u is consistent with a* discrete importance tradeoff statement $H \implies G \unrhd_{\beta}^{\alpha} rG'$ *iff all of the following conditions hold. G is mutually utility independent of $\overline{G}$, G' is mutually utility independent of $\overline{G'}$, and for any two baskets x and x' over G with $v(x') - v(x) \geq \alpha$, and for any two baskets y and y' over G' with $v(y') - v(y) \leq \beta$, and for all complete baskets a with $v(a) \in [H]$ we have*

$$u_G(v(a[x'])) - u_G(v(a[x])) \geq r(u_{G'}(v(a[y'])) - u_{G'}(v(a[y]))). \tag{7.5}$$

We observe that these tradeoff ratios can be normalized by requiring that both α and β are 1. The necessary variability can be expressed in r, *e.g.*, by choosing a new constant r' and setting $r' = \frac{r\beta}{\alpha}$.

This normalization results in a new tradeoff statement $H \implies G \unrhd_{\beta}^{\alpha} rG'$ which is true iff $H \implies G \unrhd_{1}^{1} \frac{r\beta}{\alpha} G'$ is true. Thus we sometimes write *normalized* discrete tradeoff statements without reference to α and β, and instead subscript with D for "discrete" like so: $H \implies G \unrhd_D rG'$, in such cases $\alpha = \beta = 1$.

This definition allows a correspondence between the discrete case and the continuous case.

Theorem 7.4 (Continuous Differences) *Suppose u is a continuous utility function over continuous attributes A, linear in each attribute. If $u \in [H \implies G \trianglerighteq_{\beta}^{\alpha} rG']$ then $u \in [H \implies G \trianglerighteq rG']$ in $\mathcal{L}(A)$.*

Proof. We have baskets x, x' over G such that $v(x') - v(x) \geq 1$, and baskets y, y' over G' such that $v(y') - v(y) \leq 1$. Under these conditions equation (7.5) holds. By linearity and continuity of u, u has constant partial derivatives, and so

$$u_G(v(a[x'])) - u_G(v(a[x])) = \sum_{i \in G} (x'(i) - x(i)) \left| \frac{\partial u}{\partial i}(v(a[x])) \right|. \tag{7.6}$$

Combining equation 7.6 and inequality 7.5 gives

$$abs(\nabla u(v(a))) \cdot (v(x') - v(x)) \geq r abs(\nabla u(v(a))) \cdot (v(y') - v(y)).$$

Then note that the above holds when $x'(i) - x(i) = 1$ and $y'(i) - y(i) = 1$. We let $\vec{p}$ and $\vec{q}$ be characteristic vectors for G, G', respectively. We make use of these substitutions in the above inequality and obtain

$$abs(\nabla u(v(x))) \cdot \vec{p} \geq r abs(\nabla u(v(y))) \cdot \vec{q}.$$

the definition of $u \in [H \implies G \trianglerighteq rG']$. $\square$

Note that in the above the requirement that u is linear is necessary. For example, if the partial derivatives of u with respect to G are nonconstant, equation 7.6 does not hold.

When utility functions are linear, there is a correspondence between this definition of discrete importance and the definition of discrete preference given earlier.

Theorem 7.5 *Suppose we have two tradeoff statements, $S = H \implies G \trianglerighteq_D rG'$, and $S' = H \implies d \succsim rd'$. Then let $\vec{p}, \vec{q}$ be characteristic vectors for G, G' respectively, and let d be a basket with $v(d) = \vec{p}$, and d' be a basket with $v(d') = \vec{q}$. The for a utility function u linear in the attributes A, if $u \in [S]$ then $u \in [S']$.*

Proof. If u is consistent with S then at all points $\vec{a} \in [H]$ we have

$$u(v(a[x'])) - u(v(a[x])) \geq r(u(v(a[y'])) - u(v(a[y']))) \tag{7.7}$$

when x, x' are baskets over G such that $v(x') - v(x) \geq 1$ and similarly y, y' are baskets over G' with $v(y') - v(y) \leq 1$. Since u is linear, it follows that G is utility independent of $\overline{G}$, and similarly G' is utility independent of $\overline{G'}$. This implies that u is of the form $u = u_G(v(a)) + u_{G'}(v(a)) + u_{\overline{G \cup G'}}(v(a))$, and inequality (7.7) is equivalent to:

$$u_G(v(a[x'])) - u_G(v(a[x])) \geq r(u_{G'}(v(a[y'])) - u_{G'}(v(a[y]))). \tag{7.8}$$

Again, since u is linear, the partial derivatives are merely the slopes of the function. Thus, using two arbitrary complete market baskets b, b', we have

$$u_G(v(a[x'])) - u_G(v(a[x])) = \sum_{i \in G} \frac{u(v(b[(i = b'(i))])) - u(v(b))}{b'(i) - b(i)}(x'_i - x_i). \tag{7.9}$$

Note that the above holds when $x'_i - x_i \geq 1$ and $y'_i - y_i \leq 1$. If we substitute 1 for each of these differences, we can write inequality (7.8) using substitution (7.9) and characteristic vectors $\vec{p}, \vec{q}$ for G, G'. Thus we obtain

$$\Delta u(b, b') \cdot \vec{p} \geq r \Delta u(b, b') \cdot \vec{q}.$$

This is the condition required for $u \in [S']$. $\square$

7.2.2 *Ceteris Paribus* Preferences over Binary Attributes

An important special case of the language for preferences over discrete attributes is the case of preferences between two binary variables. This is the most basic tradeoff preference that can be stated with the formalism given in this chapter.

Let us assume that P, Q are binary variables and someone wishes to state that $P \succsim rQ$, other things being equal, or *ceteris paribus*, as described in chapter 5. This can be loosely interpreted as meaning that the utility increases in the P-direction r times faster than it increases in the Q-direction, just as in the usual case of discrete tradeoff preferences. In the binary case, however, it is more explicit to say that the preference for P over Q is expressing a belief that the utility increases in the *positive-P* direction r times faster than in the *positive-Q* direction. Thus the important parts of the following definition are the choices of market baskets x and y.

Definition 7.3 *Given binary attributes P, Q, and scalar $r \geq 1$, let x, y be market baskets with $x = \{(P = \rho(p)), (Q = \rho(\neg q)))\}$, and $y = \{(P = \rho(\neg p)), (Q = \rho(q))\}$, then u is consistent with a binary* ceteris paribus *comparison $H \implies P \trianglerighteq_{CP} rQ$ in $\mathcal{L}(A)$, if and only if*

$$\Delta u(v(z[x]), v(z[y])) \cdot v(x) \geq r \Delta u(v(z[y]), v(z[x])) \cdot v(y)$$

at all complete market baskets z such that $v(z) \in [H]$.

We can show that the definition given for binary *ceteris paribus* comparisons respects the usual *ceteris paribus* condition on the utility function. Note that when P and Q are binary attributes, u is by definition linear in P and in Q.

Theorem 7.6 *If P, Q are binary attributes, x, y are market baskets with $x = \{(P = \rho(p)), (Q = \rho(\neg q)))\}$, and $y = \{(P = \rho(\neg p)), (Q = \rho(q))\}$, then a binary* ceteris paribus *comparison $H \implies P \trianglerighteq_{CP} rQ$ implies*

$$u(v(z[x])) \geq u(v(z[y])), \tag{7.10}$$

at all complete market baskets z such that $v(z) \in [H]$.

Proof. The definition of binary tradeoffs expands to

$$\begin{gathered}
(u(v(z[\{(P=\rho(p)),(Q=\rho(\neg q))\}]))-u(v(z[\{(P=\rho(\neg p)),(Q=\rho(\neg q))\}])))/ \\
(\rho(p)-\rho(\neg p))\rho(p)+ \\
(u(v(z[\{(P=\rho(p)),(Q=\rho(\neg q))\}]))-u(v(z[\{(P=\rho(p)),(Q=\rho(q))\}])))/ \\
(\rho(\neg q)-\rho(q))\rho(\neg q) \\
\geq \\
r((u(v(z[\{(P=\rho(\neg p)),(Q=\rho(\neg q))\}]))-u(v(z[\{(P=\rho(p)),(Q=\rho(\neg q))\}])))/ \\
(\rho(\neg p)-\rho(p))\rho(\neg p)+ \\
(u(v(z[\{(P=\rho(p)),(Q=\rho(q))\}]))-u(v(z[\{(P=\rho(p)),(Q=\rho(\neg q))\}])))/ \\
(\rho(q)-\rho(\neg q))\rho(q)).
\end{gathered}$$

This simplifies to

$$\begin{gathered}
(u(v(z[\{(P=\rho(p))\}]))-u(v(z[\{(P=\rho(\neg p))\}])))*(\rho(p)-r\rho(\neg p))/(\rho(p)-\rho(\neg p))\geq \\
(u(v(z[\{(Q=\rho(q))\}]))-u(v(z[\{(Q=\rho(\neg q))\}])))*(r\rho(q)-\rho(\neg q))/(\rho(q)-\rho(\neg q)).
\end{gathered}$$

The fractional terms in the above inequality are simply the slope of the utility function in a particular dimension. When u is linear, this slope is constant, and without loss of generality let t_p be the slope in the P-dimension, and t_q be the slope in the Q-dimension. Then the above is simply

$$t_p(\rho(p)-r\rho(\neg p))\geq t_q(r\rho(q)-\rho(\neg q)).$$

We can rearrange and add a constant to get:

$$k+t_p\rho(p)+t_q\rho(\neg q)\geq k+r(t_p\rho(\neg p)+t_q\rho(q)),$$

where we let k represent the contribution to utility by the attributes other than P, Q. By the *ceteris paribus* clause of the statement, this contribution remains constant. Thus, we have

$$u(v(z[x]))\geq u(v(z[y]))+(r-1)(t_p\rho(\neg p)+t_q\rho(q)).$$

We can assume that $\rho(\neg p)$ and $\rho(q)$ are positive, without loss of generality, since ρ is an arbitrary isomorphism. For $r\geq 1$, we have

$$u(v(z[x]))\geq u(v(z[y])),$$

as required. □

This theorem shows that the partial derivative semantics given here are sufficient, in some cases, to recover the semantics of *ceteris paribus* preferences given earlier in the thesis. That is, a binary tradeoff statement $P \trianglerighteq_{CP} Q$ implies a statement of preference $(P=p) \succsim_{CP} (Q=q)$. Recall that, in language $\mathcal{L}(A)$, $(P=p) \succsim_{CP} (Q=q)$ if and only if for all pairs of baskets b, b' with $b[\{P\}\cup\{Q\}]=b'[\{P\}\cup\{Q\}]$, we have $b \succsim b'$. And of course, when a utility function is involved, this is equivalent to $u(b)\geq u(b')$.

Theorem 7.7 *For binary attributes P, Q, a* ceteris paribus*binary tradeoff statement $P \trianglerighteq_{CP} Q$ is equivalent to a qualitative* ceteris paribus*preference statement $(P=p) \succsim_{CP} (Q=q)$.*

This follows directly from theorem 7.6.

7.3 No Attribute Interactions

There are some kinds of statements a user might wish to make that our formalism cannot cleanly handle. Suppose a user is buying a computer. The decision maker might wish to state that "Price and warranty together are r-times more important than the price or warranty individually." It captures the desire of the user to balance, somehow, the price of the computer with the type of warranty provided. These two together are more important than either individually.

Let us suppose that the decision maker encodes this sentiment as two importance statements: firstly, $\{price,\ warranty\} \rhd r\ \{price\}$, and secondly, $\{price,\ warranty\} \rhd r\ \{warranty\}$. Let us assume for simplicity that in this case $A = \{price,\ warranty\}$. Thus we have

$$abs(\nabla u(\vec{a})) \cdot \langle 1, 1\rangle > r(abs(\nabla u(\vec{a})) \cdot \langle 1, 0\rangle)$$

and

$$abs(\nabla u(\vec{a})) \cdot \langle 1, 1\rangle > r(abs(\nabla u(\vec{a})) \cdot \langle 0, 1\rangle).$$

By definition 7.1 we have:

$$\frac{\partial u}{\partial p}(\vec{a}) + \frac{\partial u}{\partial w}(\vec{a}) > r\frac{\partial u}{\partial p}(\vec{a}),$$

and this when taken together with

$$\frac{\partial u}{\partial p}(\vec{a}) + \frac{\partial u}{\partial w}(\vec{a}) > r\frac{\partial u}{\partial w}(\vec{a}),$$

implies that

$$\tfrac{\partial u}{\partial w}(\vec{a}) > (r-1)\tfrac{\partial u}{\partial p}(\vec{a}), \quad \tfrac{\partial u}{\partial p}(\vec{a}) > (r-1)\tfrac{\partial u}{\partial w}(\vec{a}),$$

which are both true only for $r < 2$; and these are satisfied regardless of any values of the partials of *price* and *warranty.* For values of $r \geq 2$, these two statements cannot be simultaneously true. Thus we have a pair of statements, $\{price,\ warranty\} \rhd r\ \{price\}$, and $\{price,\ warranty\} \rhd r\ \{warranty\}$, that are either tautological or contradictory; we must conclude that these types of statements, if presented, are of little value in specifying preferences.

Chapter 8

Accommodating Exceptions

Our aim in this chapter is to capture a natural intuitive assumption about the nature of generalizations and special cases. In everyday reasoning and communication, it is normal to make equivocal generalizations that cover broad categories or situations, and then to make refinements by overriding the generalizations with more specific statements, holding in more specific categories and situations. Suppose I state that for automobiles, the fuel efficiency is more important to me than the safety record of the car. However, in the special case of SUVs, this case is reversed and the safety is more important to me than the milage. These two statements, taken together, constitute a generalization and an exception to the generalization, because SUVs are a subset of automobiles. This chapter is devoted to systematizing the reasoning behind this example.

We allow conditional tradeoff statements to be made that conflict with other statements. In cases where one statement is a refinement or specification of another, the more specific statement will take priority. This allows a decision maker to provide generalizations and exceptions for special cases of their preferences.

By way of an important aside, we remark that we do not generally expect decision makers to exhibit perfect consistency with their preferences, either in their expression or in their underlying structure. Our interpretation of generalizations and exceptions solves certain types of "conflicts" between preferences, that may arise in elicitation, but other types of conflicts between preference statements cannot be given such a clean interpretation. We merely try to allow or accommodate whatever modest type of inconsistency that we can.

8.1 Generalization and Refinement

We assume that we are given conditional tradeoff statements in $\mathcal{L}(A)_1$ or $\mathcal{L}(A)_2$ that are either conditional attribute tradeoff statements or conditional basket tradeoff statements. For simplicity of exposition, we will assume we are dealing with conditional basket tradeoff statements, although what we discuss applies equally to conditional attribute tradeoff statements. Recall that a conditional basket tradeoff is of the form $S = H \implies T$, where $[H] \subseteq \vec{A}$ is a subspace where the basket tradeoff

T holds, which we will refer to as the *extent* of the preference. Given two tradeoff statements $S_1 = H \implies T_1$ and $S_2 = H' \implies T_2$, with $[H'] \subseteq [H]$, we expect that T_1 holds in $[H] \backslash [H']$ and T_2 holds in $[H']$. Since the extent of S_2 is smaller and a subset of the extent of S_1, S_2 is considered to be more specific. In this case, S_1 is treated as a general statement and S_2 is its refinement or exception. We use *refinement* when S_1 and S_2 are not mutually exclusive and *exception* when S_1 and S_2 cannot both hold simultaneously. To make this explicit we provide definitions of exception and refinement.

Definition 8.1 (Generalization and Exception) *Given two conditional basket statements $S_1 = H \implies T_1$, $S_2 = H' \implies T_2$, if $[H'] \subset [H]$ and $\{S_1\} \cup \{S_2\}$ is unsatisfiable, then S_2 is said to be an* exception to S_1.

Definition 8.2 (Generalization and Refinement) *Given two conditional basket statements $S_1 = H \implies T_1$, $S_2 = H' \implies T_2$, if $[H'] \subset [H]$ and $\{S_1\} \cup \{S_2\}$ is satisfiable, then S_2 is said to be a* refinement of S_1.

The above will cover simple kinds of generalizations. Clearly there are more complicated scenarios. Continuing the exposition, when S_2 is an exception to S_1, S_2 can itself have exceptions. $[H]$ and $[H']$ might intersect without one being a subset of the other. There could, in general, be many nested and intersecting preferences and exceptions.

The problem becomes: given some set of statements M, which of these statements hold at which points or regions in the space $\vec{A}$? We will call any particular set of consistent statements that hold at a point or region an *interpretation* for that point or region.

8.2 Interpretations of Overlapping Statements

Given some set of preference statements M in $\mathcal{L}(A)_1$ or $\mathcal{L}(A)_2$, which of these preferences hold at which points or regions in the space $\vec{A}$?

First consider any point $\vec{x}$ in $\vec{A}$. There is a region or neighborhood around that point such that all points in this region fall into the extent of the same set of preferences. For a set of preference statements M, we define an equivalence relation on $\vec{A}$, and equivalence class $[\vec{x}]_M$ such that $\vec{y} \in [\vec{x}]_M$ iff for each preference $S = H \implies T$, $S \in M$, if $\vec{x} \in [H]$, we have $\vec{y} \in [H]$. We denote the characteristic set of preferences for $[\vec{x}]_M$ by $M(\vec{x})$.

Next, given statements M and an equivalence class $[\vec{x}]_M$, we wish to find a consistent interpretation for $[\vec{x}]_M$. While any consistent interpretation is a possible solution, we want interpretations that respect the exceptions and refinements to wider preferences, as we have described them above. Loosely speaking then, the preferences that hold on $[\vec{x}]_M$ are those that include $[\vec{x}]_M$ in their extent, do not have exceptions, and that do not conflict with other such preferences.

Given a set of preference statements M, $R \subseteq M$ is a *maximal consistent* subset of M when for any $S \in M \backslash R$, $R \cup \{S\}$ is inconsistent. A maximal consistent set is then a set of conditional tradeoff preference that cannot have another preference added to it without becoming inconsistent. This also means exceptions to existing preference statements cannot be added to R, since a statement and an exception to it are inconsistent. However a maximal consistent interpretation may end up including a preference that has an exception; this is not the intention behind the exception and generalization heuristic, so we must modify this definition.

Borrowing terminology from the non-monotonic reasoning literature, a *credulous* interpretation for $[\vec{x}]_M$ is a maximal consistent subset R of M such that no $S \in R$ has an exception $S' \in M$. A credulous interpretation does not include a statement that has been overridden by an exception; it will only include the exception. Note that there are many possible credulous interpretations for $[\vec{x}]_M$. The intersection of all the credulous interpretations is itself an interpretation, and we call it the *skeptical* interpretation.

Note that in both skeptical and credulous interpretations no preference and its exception hold simultaneously. This means both types of interpretation agree with our intuition about generalization and exception. The difference here is how skeptical and credulous interpretations behave when there are conflicting preferences without clear heuristics to determine which of those to believe. The skeptical interpretation will believe none of the conflicting preferences, while the credulous interpretations will believe some consistent subset of the conflicting preferences.

We describe an algorithm, *Skep-Interp* in figure 8.2, that computes the skeptical interpretation of $[\vec{x}]_M$ given $\vec{x}$. This algorithm proceeds as follows. First it checks all pairs of preferences in $M(\vec{x})$. This algorithm keeps two lists, one of acceptable preferences R and one of unacceptable preferences U. Then *Skep-Interp* checks each pair of preferences for consistency. The only case when conflicting preferences are allowed is when one is the exception to the other, and in such cases, only the exception is allowed.

Theorem 8.1 (Skeptical Interpretation) *Algorithm* Skep-Interp *produces the skeptical interpretation for* $[\vec{x}]_M$.

This theorem holds since all preference pairs are considered, and only preferences that are consistent are included in the interpretation. This is all that is required for the definition of a skeptical interpretation. The interpretation is maximal since every pair of preferences is considered. Any preference not included in R necessarily conflicts with some other preference.

An algorithm for a credulous interpretation can be obtained from *Skep-Interp* by changing step 2.b.iii) to choose one of S_1, S_2 arbitrarily and then add the chosen one to R, provided it is not already in U.

8.3 Comments about Exceptions

The interpretations in this section are meant to be more of an illustration of possible solutions to the problem of exceptions and generalizations. There may well be other types of nonmonotonic reasoning techniques that can be included here. And those

Skep-Interp

Input: Set of tradeoff preferences M in $\mathcal{L}(A)_1$ or $\mathcal{L}(A)_2$, a point $\vec{x} \in \vec{A}$ or equivalence class $[\vec{x}]_M$

Output: Set of tradeoffs preferences R that hold at $[\vec{x}]_M$.

1. $R \leftarrow \emptyset$, $U \leftarrow \emptyset$.
2. For each pair $\{S_1, S_2\} \in M(\vec{x})$ do
   (a) If $\{S_1, S_2\}$ is satisfiable then
      i. if $S_1 \notin U$ then let $R \leftarrow R \cup \{S_1\}$,
      ii. if $S_2 \notin U$ then let $R \leftarrow R \cup \{S_2\}$.
   (b) If $\{S_1, S_2\}$ is unsatisfiable then
      i. if S_1 is an exception to S_2 and $S_1 \notin U$ then let $R \leftarrow R \cup \{S_1\}$ and let $U \leftarrow U \cup \{S_2\}$,
      ii. if S_2 is an exception to S_1 and $S_2 \notin U$ then let $R \leftarrow R \cup \{S_2\}$ and let $U \leftarrow U \cup \{S_1\}$.
      iii. if neither S_1 or S_2 is an exception to the other, then let $U \leftarrow U \cup \{S_1, S_2\}$ and let $R \leftarrow R \backslash \{S_1, S_2\}$.
3. Output R.

Figure 8-1: Pseudo-code listing for algorithm Skep-Interp.

might well address intuitions other than the exception/generalization system here, based upon the extent of preferences. We do not go into more detail here, as this is more properly a topic for the nonmonotonic reasoning literature.

Chapter 9

Utility Function Construction

We have given many representations of different types of preferences. It remains now to join them together by providing utility functions that compute partial orderings over the attribute space consistent with a given set of preferences.

In this chapter, we are concerned to find a utility function that represents whatever preferences a user specifies. We describe utility functions that represent any given satisfiable set of statements of the main types described in in $\mathcal{L}(A), \mathcal{L}(A)_1, \mathcal{L}(A)_2$: conditional qualitative *ceteris paribus*preference (see section 5.1); conditional basket tradeoffs (see section 6.1); conditional attribute tradeoff statements (see section 7.1).

Thus, we address these questions at present: Given however many or few preferences a user wishes to state, can we find a utility function consistent with them? Under what circumstances can we find one efficiently? Does this provide a unified and flexible framework for the expression of various types of preferences?

Much of this chapter is an augmentation of techniques presented in [MD04] for constructing an ordinal utility function for *ceteris paribus*preferences over binary attributes. The presentation given here is updated to accommodate all of the types of preferences discussed so far. There are still some sections and results that require little modification from the original, and as such are skipped here. For the modified theorems we present here as analogues to the theorems of [MD04], we relegate their proofs to an appendix when there are no substantive changes to these proofs.

9.1 Approach

In broad terms, we generate an ordinal utility function from sets M of qualitative *ceteris paribus*preferences, M' of conditional basket tradeoffs, and M'' of conditional attribute tradeoffs, over attributes A in the following steps.

First we examine the qualitative *ceteris paribus*preferences stated in M and determine which attributes might be assumed utility independent of which other attributes. Utility independence provides computational benefits, since utility independent sets of attributes can be considered without regard to the other attributes. Next, again using the qualitative *ceteris paribus*preferences in M, we can define subutility functions for each utility independent set of attributes. Such methods are based on representing the preorders consistent with the preferences M by building a graph over assignments to the attributes. These methods are unchanged from the presentation in [MD04], and are not repeated herein. Finally, to assign relative weights of satisfaction of different attributes, we consider the tradeoff and attribute preferences in M' and M'', and construct and solve a linear programming problem. In the end, we have built a utility function u that represents $M \cup M' \cup M''$ and can be used to quickly evaluate the utility of different assignments to values of A.

9.2 Generalized Additive Independence

In previous work [MD04], we presented an algorithm for computing a generalized additive independent partition of the attributes by looking at characteristics of the input preferences, when those preferences are *ceteris paribus*statements over binary attributes. Although our preferences development here was over discrete attributes with finite domains, the methodology presented in that source can be used largely unchanged.

9.2.1 Qualitative *Ceteris Paribus* Preferences and Utility Independence

The condition for generalized utility independence in implication (2.1) is stronger than we require. Instead of satisfying implication (2.1), we would rather assume that all attributes are utility independent of all other attributes and then look for cases where this assumption is explicitly controverted by the user's preferences. With this approach in mind, we present methods to determine when attributes must be utility dependent.

We first look at each pairs of qualitative *ceteris paribus*preference statements and check if the preference expressed over one attribute changes with the value of another attribute. This can sometimes be read from the preferences themselves. Consider an example. If we have two preferences

$$\begin{aligned} S_1 = H &\implies (\alpha_1 = w_1) \wedge (\alpha_2 = w_3) \succ_{CP} (\alpha_1 = w_2) \wedge (\alpha_2 = w_3), \\ S_2 = H &\implies (\alpha_1 = w_2) \wedge (\alpha_2 = w_4) \succ_{CP} (\alpha_1 = w_1) \wedge (\alpha_2 = w_4), \end{aligned}$$

then it is evident that the preference for attribute α_1 cannot be utility independent from α_2, since both of the stances that $(\alpha_1 = w_1) \succ_{CP} (\alpha_1 = w_2)$ and $(\alpha_1 = w_2) \succ_{CP} (\alpha_1 = w_1)$ cannot hold simultaneously. The remaining conclusion is that α_1 is utility dependent on α_2, and that when $\alpha_2 = w_3$ we have

$$(\alpha_1 = w_1) \succ_{CP} (\alpha_1 = w_2),$$

and when $\alpha_2 = w_4$ we have

$$(\alpha_1 = w_2) \succ_{CP} (\alpha_1 = w_1).$$

Following this intuition, a simple algorithm can check each pair of preferences to see if they display this sort of obvious utility dependence. In [MD04], we present an algorithm that does so called *UIDecomposition*. This algorithm requires no other input besides the preference statements themselves, and is sound but not complete. There may yet be utility dependence latent in the preferences that it does not discover. In [MD04], we argue that any complete algorithm for finding utility dependence must be of worst-case exponential complexity in the number of attributes. Thus, while a complete version of the *UIDecomposition* function is possible in principle, it is not practical.

UIDecomposition outputs two sets of sets of attributes. It computes a partition C of A into utility-independent attribute sets and an associated set of attribute sets D such that each attribute set $C_i \in C$ is utility dependent upon a (possibly empty) attribute set $D_i \in D$.

We present an updated version of the theorem in [MD04] that describes conditions when *UIDecomposition* is guaranteed to find all utility dependence extant in a set of preferences T.

Theorem 9.1 (Independence Construction) *If the input preferences T are such that each $S \in T$ is a conditional qualitative* ceteris paribus*preference $S = H \implies b \succ_{CP} b'$ where b, b' are baskets over A, and if for two baskets x, y over A we have $u(x) > u(y)$ for all $u \in [T]$, then there exists $S' \in T$ such that $S' = H \implies x \succ_{CP} y$, then the* UIDecomposition *function computes a partition C of the attributes A such that each set $C_i \in C$ is utility independent of the attributes in $A \backslash \{C_i \cup D_i\}$, and further, for each C_i, no set $D'_i \subset D_i$ exists such that C_i is utility independent of $A \backslash D'_i$.*

When this condition does not hold, utility dependence undetected by function *UIDecomposition* can cause an error in a later step of the algorithm, but we describe heuristics that correct the problem in many cases (see Section 9.3.6).

This section has covered the cases of conditional qualitative *ceteris paribus*preferences, but it remains to consider what tradeoff preferences can tell us about utility dependence in the domain.

9.2.2 Tradeoff Preferences and Utility Independence

We have just reviewed a technique where the structure of qualitative *ceteris paribus*preference statements is used to determine which variables are known to be utility dependent. This allows us to make intelligent assumptions about the structure of the utility functions that are consistent with the input preferences. Unfortunately, we show here that no such methods are possible for tradeoff preferences. The good news is that no such methods are necessary.

We examine two concerns in this section. Firstly, does a tradeoff statement between attributes X and attributes Y imply that X and Y are utility independent? The answer is no. Secondly, is it *possible* that X and Y are utility independent? The answer is yes whenever the tradeoffs made between X and Y are satisfiable.

It is not true that every tradeoff statement means there must be utility independence between the related attributes. We present this result by demonstrating a counterexample.

Theorem 9.2 (Utility Dependence) *There exists some basket tradeoff statement S over some domain A such that there exists $u \in [S]$ with the attributes related in S utility dependent.*

Proof. Consider the following example. Let $A = \{X, Y\}$. Let $S = b \succ rb'$ for baskets $b = (X = 1), b' = (Y = 1)$, be a basket tradeoff statement. Then consider a utility function: $u(x, y) = (rx + y - 1)^2$ that exhibits both utility dependence of X on Y and obeys the tradeoff statement S.

The statement $b \succ rb'$ implies this constraint on utility functions:

$$\frac{\partial u}{\partial X}(x, y) \geq r \frac{\partial u}{\partial Y}(x, y)$$

for all possible x, y. We now show that $u(x, y) = (rx + y - 1)^2$ satisfies this constraint. By simple algebra,

$$u(x, y) = r^2x^2 + 2rxy - 2rx + y^2 - 2y + 1.$$

Then we have the partial derivatives, as follows

$$\begin{array}{l} \frac{\partial u}{\partial X}(x, y) = 2r^2x + 2ry - 2r \\ \frac{\partial u}{\partial Y}(x, y) = 2y + 2rx - 2 \end{array}$$

Using the above we can now verify that the main condition holds.

$$2r^2x + 2ry - 2r \geq r(2y + 2rx - 2)$$

simplifies to

$$2r^2x + 2ry - 2r \geq 2ry + 2r^2x - 2r$$

and this holds for all pairs x, y.

It remains to show that u exhibits utility dependence of X on Y. We do this by observing that $u(0, 0) > u(1, 0)$ but that $u(1, 1) > u(0, 1)$. This shows that the preference for $X = 0$ over $X = 1$ holds when $Y = 0$ but reverses when $Y = 1$. This is the definition of utility dependence. $\square$

The opposite concern is also interesting. Is it always possible to create a linear additive utility function (and therefore one that exhibits utility independence) given any set of tradeoff preferences? We will show that it is possible to construct a piecewise linear utility function for any set of satisfiable preferences.

Theorem 9.3 (Utility Independence) *For any set of unconditional basket tradeoff statements T, if T is satisfiable, then there exists $u \in [T]$ such that u is linear in each attribute in A.*

Proof. We are given some set of unconditional basket tradeoff preferences T over some set of attributes A, these are of the form $b \succ rb'$. These tradeoff statements, in turn, require some conditions C of the partial derivatives of the utility function, of the form

$$\nabla u(\vec{a}) \cdot v(b) \geq r \nabla u(\vec{a}) \cdot v(b')$$

for all $\vec{a}, \vec{a}' \in \vec{A}$ and for some particular b, b'. These constraints C hold at all points $\vec{a}$ in the preference space. A solution to C is a value for $\nabla u(\vec{a})$. Since constraints C hold at all $\vec{a}$, then there is no need to worry about different solutions at different points; any solution to C satisfies C at all points $\vec{a}$. The solution to constraints C is a vector of numbers, let it be $\vec{w}$, and this vector is the vector of partial derivatives $\nabla u(\vec{a})$. Therefore, there exists u with $\nabla u(\vec{a}) = \vec{w}$ for all $\vec{a} \in \vec{A}$. This function u satisfies the condition, and proves the theorem. □

We can extend this result to conditional tradeoff statements, by considering piecewise linear utility functions; essentially a different linear utility function is required for each separate region indicated by the conditions of the tradeoff preferences. We defer this discussion for the time being.

9.3 Utility Construction

We now describe how to compute one utility function consistent with the sets of input preferences, M, M', M''. We take the partition of $A : C' = \{C'_1, C'_2, \ldots, C'_Q\}$ discussed in the previous section, and the corresponding set of sets of attributes $D' = \{D'_1, D'_2, \ldots, D'_Q\}$, where each set of attributes C'_i is utility dependent on the attributes D'_i and utility independent of $A \backslash D'_i$. We define a set of sets $C = \{C_1, C_2, ..., C_Q\}$ such that $C_i = (C'_i \cup D'_i)$. We will look for an additive utility function that is a linear combination of subutility functions u_i, each subutility a separate function of a particular C_i. We associate a *scaling parameter* t_i with each u_i such that the utility of a model is

$$u(m) = \sum_{i=1}^{Q} t_i u_i(m). \tag{9.1}$$

We refer to this as a *generalized additive utility function.* We will argue that this is consistent with sets of preferences M, M', M'' where M is a set of qualitative *ceteris paribus*preferences statements in $\mathcal{L}(A)$, M' a set of conditional basket tradeoff statements in $\mathcal{L}(A)_1$, and M'' a set of conditional attribute tradeoffs in $\mathcal{L}(A)_2$. We have two tasks: to craft the subutility functions u_i, and to choose the scaling constants t_i. We will accomplish these two tasks in roughly the following way. We will show how a preference can be *restricted* to a set C_i. This is essentially a shortening of a preference to a particular set of attributes. By restricting preferences to the sets C_i, we can use these shortened forms of the preferences to make graphical utility functions for the partial utility functions u_i. Preferences that are consistent in general can be locally inconsistent when restricted to different attribute sets. These inconsistencies can be phrased as constraints and resolved using a boolean constraint satisfaction solver (SAT). To assign values to scaling parameters t_i, we will define a set of linear inequalities which constrain the variables t_i. The linear inequalities can then be solved using standard methods for solving linear programming problems. The solutions to the inequalities are the values for the scaling parameters t_i. Along the way we will show some helpful heuristics.

We first describe the subutility functions, and then their linear combination into a full utility function.

9.3.1 Subutility Functions

Our task is to define subutility functions $u_{C_i}(\vec{a})$ that take as input the values for the attributes A, and return an integer. We define such a function relative to some set if preference $S \in M$. We say that a subutility function u_{C_i} is *ϵ-consistent* with M if either $u_{C_i}(\vec{a}) \geq \epsilon + u_{C_i}(\vec{a}')$ whenever $(a, a') \models S, \forall S \in M$, or $u_{C_i}(\vec{a}) + \epsilon \leq u_{C_i}(\vec{a}')$ whenever $(a, a') \models S, \forall S \in M$, where $\epsilon > 0$. In the following, we generally let $\epsilon = 1$ and prove results for 1-consistency. In general, it is not necessary to use $\epsilon = 1$, but using 1 will make some calculations simpler.

We define the *restriction* of a basket b to a set of attributes C, written $b \upharpoonright C$. The restriction of a basket b to C is another basket b', but over fewer attributes. We replace all assignments in b to attributes outside of C by assignments to $\perp$. This, in effect, deletes the elements of the basket outside of C. Similarly, we can define the restriction of a qualitative *ceteris paribus*preference statement S to C by

$$S \upharpoonright C = (p \upharpoonright C) \succ_{CP} (q \upharpoonright C).$$

We can now show that 1-consistency implies the consistency of u with S.

Theorem 9.4 (Subutilities) *Given qualitative* ceteris paribus*preferences M and a mutually utility-independent partition C of A, if each u_i is 1-consistent with M, then*

$$u(\vec{a}) = \sum_{i=1}^{Q} t_i u_i(\vec{a})$$

with $t_i(u_i(b) - u_i(b')) > 0$ when $b \succ_{CP} b'$, is consistent with $S \in M$.

Proof. Given in the appendix.

In this theorem we assume that $t_i(u_i(b) - u_i(b')) > 0$ when $b \succ_{CP} b'$, which amounts to an assumption about the way the domains of attributes are constructed. We assume, here and in the following, that if attributes are increasing in utility when they increase in quantity, that is, when $u_i(b) - u_i(b') > 0$ when $b \succ b'$, then $t_i > 0$, which means an increase in the amount of a good thing increases total utility. In these cases we say that u_i is *positively ordered.* Conversely, we assume that if an attribute is decreasing in utility as it increases (such as cost or the length of a delay), and therefore $u_i(b) - u_i(b') < 0$ when $b \succ b'$, then we have $t_i < 0$; we say that u_i is *negatively ordered.* These are not very restrictive assumptions, and are more general than merely assuming $t_i > 0$ in all cases. Further, we can frequently assume that all subutility functions are in fact positively ordered.

We show how to generate a 1-consistent subutility function by constructing a *restricted model graph,* $G_i(R)$ for a set of preferences R in $\mathcal{L}(A)$. $G_i(R)$ is a multi-graph with directed edges, where each basket b over R is a node in the graph. Let b_1, b_2 be baskets over C_i with $b_1 \neq b_2$. There exists one directed edge in the restricted model graph $G_i(R)$ from node b_1 to node b_2 for each distinct preference $S \in R$ such that $(b_1, b_2) \models S$, where $b_1 \neq b_2$. For example, if $(b_1, b_2) \models S_1$ and $(b_1, b_2) \models S_2$, there exists two edges from node b_1 to b_2. We label the edges with the preference S that causes the edge. The interpretation of an edge $e(b_1, b_2)$ in $G_i(C_i)$ is of a strict preference for b_1 over b_2. The construction of this graph $G_i(R)$ parallels the construction of the general model graph G, as described in [MD04].

With a restricted model graph $G_i(R)$ defined, there is still the matter of defining a utility function on this graph. It appears we could choose any of the Ordinal Utility Functions described in [MD04] to be the utility function for u_i, and preserve further results in this section. For example, we could use the Minimizing Graphical Utility Function for u_i. Recall that in this ordering, each node has utility equal to the length of the longest path originating at the node. If we use the Minimizing Graphical Utility Function, then we want $u_i(b)$ to return the length of the longest path starting at node b in graph $G_i(R)$.

Lemma 9.3.1 (Cycle-Free Subutility) *Given a set of preferences $R \subseteq M$ and a set of attributes $C_i \subseteq A$ such that the restricted model graph $G_i(R)$ is cycle-free, and $u_i(\vec{a})$ is the minimizing graphical utility function over $G_i(R)$, then subutility function $u_i(\vec{a})$ for C_i is 1-consistent with R.*

Proof. Given in the appendix.

For further discussions of these subutility functions, the interested reader is referred to [MD04].

9.3.2 Conflicting Preferences

Although we assume that the input preferences M are all strict preferences, and have no conflicting preferences, it is possible that a restricted model graph $G_i(R)$ for $C_i \subset A$ has strict-edge cycles in it even though a model graph $G(A)$ for A does not. Since we are creating subutility functions for use in a generalized additive utility function, we cannot use cycles. Using cycles would break many of the heuristics we will define.

Consider a set of preferences $R \subseteq M$, and set C of attribute sets $C_i \subseteq A$. The preferences R *conflict* if there exists some attribute set $C_i \in C$ such that no subutility function u_i for C_i can be 1-consistent with all $S \in R$. When such a attribute set C_i exists, we say that R *conflict on* C_i or that R *conflict on* u_i.

Lemma 9.3.2 (Conflict Cycles) *Given a set of strict preferences $R \subseteq A$ and a set C of attribute sets of A, the preferences R conflict if and only if the preferences R imply a cycle in any restricted model graph $G_i(R)$ of some set of attributes $C_i \in C$.*

Proof. Given in the appendix.

Note that if the preferences R imply a cycle in the (unrestricted) model graph $G(R)$ over all attributes, A, then the preferences R represent contradictory preferences.

We call a set of qualitative *ceteris paribus*preferences R_i the *relevant preference set for* C_i if R_i is the set of preferences $r \in M$ such that $(b_1, b_2) \models r \implies (b_1 \upharpoonright C_i) \neq$

$(b_2 \upharpoonright C_i)$. The preferences R_i may or may not conflict on C_i. If the preferences R_i do not conflict on C_i, we say that R_i is *conflict-free.* If some subset of the preferences R_i do conflict on C_i, we resolve conflicts by choosing a subset of preferences $\overline{R_i} \subseteq R_i$ that is conflict-free. Operationally, we will choose some of the conflicting preferences of R_i and exclude them from the set $\overline{R_i}$. How to choose which preferences to exclude from $\overline{R_i}$ is a central point of discussion in the rest of the section. For now, just note that our choice of preferences to remove influences our choice of scaling parameters t_i.

We define two properties of conflicting preference sets on R_i. Suppose there are several sets of conflicting preferences $Y_1, Y_2, ..., Y_K$, where each is composed of statements $r \in R_i$, for an attribute set C_i. A set of conflicting statements Y_k is *minimal* if for any $r \in Y_k$, $Y_k \backslash r$ implies a subset of the cycles implied by Y_k on C_i. The set of conflicting preference sets $Y = \{Y_1, Y_2, ..., Y_K\}$ is *minimal and complete* if each conflicting subset R_i' of R_i is represented by some $Y_k \in Y$ such that for any $r \in Y_k$, $Y_k \backslash r$ implies a subset of the cycles implied by R_i'.

Theorem 9.5 (Conflict-free Statement Set) *Given a relevant preference set R_i for C_i, and a minimal, complete set of conflicting preference sets Y for R_i, and*

$$\overline{R_i} = R_i \backslash \{r_1, r_2, ..., r_K\} \tag{9.2}$$

where r_j are any preferences such that $r_1 \in Y_1$, $r_2 \in Y_2$, ..., $r_K \in Y_K$, then $\overline{R_i}$ is conflict-free.

Proof. We will assume the opposite, and arrive at a contradiction. Suppose that $\overline{R_i}$ is not conflict-free, then by Lemma 9.3.2 $\overline{R_i}$ implies a cycle in $G_i(\overline{R_i})$ of C_i. Since $\overline{R_i} \subset R_i$, the same cycle is implied by R_i. By minimality and completeness of Y, there is some set of preferences $Y_j \in Y$ such that Y_j implies the cycles implied by R_i. By minimality of Y_j, if $\overline{R_i}$ implies the same cycle as Y_j, then $\overline{R_i}$ must contain the statements Y_j. But, the preferences Y_j cannot be in $\overline{R_i}$, because $r_j \in Y_j$ and we have defined $r_j \notin \overline{R_i}$ in equation 9.2. Thus, we have arrived at a contradiction, and therefore shown that $\overline{R_i}$ is conflict-free. □

Lemma 9.3.2 states that conflict sets in R are equivalent to cycles in the restricted model graph $G_i(R)$. Thus, we can find minimal sets of conflicting statements in R by using a cycle detection algorithm on the graph $G_i(R)$. We can annotate the edges in $G_i(R)$ with the preference $r \in R$ that implies the edge. This way, it is simple to translate a cycle to a set of statements. Note that if we find all cycles of $G_i(R)$, we have found a minimal and complete set of conflicting statements in R. Since one set of preferences R may cause more than one cycle in a given $G_i(R)$, there may be multiple identical conflicting sets of preferences. Thus, when translating a set of cycles to a set of sets of conflicting statements, it may be more efficient to prune away duplicate cycles, but this isn't strictly necessary.

We can construct $\overline{R_i}$ satisfying equation 9.2 given minimal conflict sets Y_k and relevant preference sets R_i. Note that there are several possible values for $\overline{R_i}$ satisfying this equation, since we can choose any $r_1, r_2, ..., r_K$ such that $r_1 \in Y_1$, $r_2 \in Y_2, ..., r_K \in Y_K$. Since r_k can be a member of more than one preference conflict set Y_k, it is possible that the cardinality of $\overline{R}_i$ varies with different choices of r_k. For purposes of computational efficiency, it is generally advantageous to have $\overline{R}_i$ have as high a cardinality as possible, but not necessary.

Given conflict-free $\overline{R}_i$, we can construct the restricted model graph $G_i(\overline{R}_i)$. This is a model graph $G_i(\overline{R}_i)$ that is cycle-free. With no cycles, we can construct a subutility function for S_i using any of the graphical utility function given in [MD04] over the restricted model graph $G_i(\overline{R}_i)$.

Theorem 9.6 (Subutility 1-Consistency) *Given a set of attributes $C_i \subseteq A$ and corresponding conflict-free relevant statement set $\overline{R}_i$, the subutility function u_i using a graphical utility function from [MD04] based on the restricted model graph $G_i(\overline{R}_i)$ is 1-consistent with all qualitative* ceteris paribus*preference statements $r \in \overline{R}_i$.*

Proof. Since $\overline{R}_i$ is conflict-free, by Lemma 9.3.2, $G_i(\overline{R}_i)$ is cycle-free. Then by Lemma 9.3.1, u_i is 1-consistent with $\overline{R}_i$. □

We call a subutility function satisfying Theorem 9.6 a *graphical subutility function for C_i*. Note that u_i may or may not be 1-consistent with $r \in (R_i \backslash \overline{R}_i)$, and in general we assume that these functions are not 1-consistent with such r. For notational convenience, we say that a subutility function u_i as described in Theorem 9.6 *agrees* with preferences $r \in \overline{R}_i$, and that u_i *disagrees* with preferences $r \in (R_i \backslash \overline{R}_i)$. 1-Consistency is implied by agreement, but not the converse, because agreement is a relation between a specific type of subutility function and a preference.

9.3.3 Consistency Condition

Recall that we have defined what it means for a utility function u to be consistent with a set of strict qualitative *ceteris paribus*preferences M as $u \in [M]$. This is equivalent to having $u(b_1) > u(b_2)$ whenever $b_1 \succ_{CP} b_2$. This translates into the following condition on the subutility functions u_i. For each statement $S \in M$ and each pair of baskets $(b_1, b_2) \models S$, we have the following:

$$\sum_{i=1}^{Q} t_i u_i(b_1) > \sum_{i=1}^{Q} t_i u_i(b_2) \tag{9.3}$$

that is, if we have a preference S implying that $b_1 \succ_{CP} b_2$, then the utility given by u to b_1 must be greater than that given to b_2. For it to be otherwise would mean our utility function u was not faithfully representing the set of preferences M. Let R_i be the relevant statement set for C_i. Examine one subutility function, u_i, and pick some $r \notin R_i$. If $(b_1, b_2) \models r$, both b_1, b_2 assign the same values to the attributes in C_i. That is, $(b_1 \upharpoonright C_i) = (b_2 \upharpoonright C_i)$. This implies $u_i(b_1) = u_i(b_2)$. So we can split the summations of equation (9.3), giving:

$$\sum_{i:r \in R_i} t_i u_i(b_1) + \sum_{i:r \notin R_i} t_i u_i(b_1) > \sum_{i:r \in R_i} t_i u_i(b_2) + \sum_{i:r \notin R_i} t_i u_i(b_2) \tag{9.4}$$

then simplify:

$$\sum_{i:r\in R_i} t_i u_i(b_1) > \sum_{i:r\in R_i} t_i u_i(b_2). \tag{9.5}$$

Given a set $C = \{C_1, ..., C_Q\}$ of subsets of A, corresponding relevant statement sets R_i for each $C_i \in C$, conflict-free preference sets $\overline{R}_i$ for each R_i, and subutility functions u_i 1-consistent with $\overline{R}_i$, define $C_a(r)$ and $C_d(r)$ as follows. For a preference r, $C_a(r)$ is the set of indices i such that $r \in \overline{R}_i$, and $C_d(r)$ is the set of indices i such that $r \in (R_i \backslash \overline{R}_i)$.

Theorem 9.7 (Preference Consistency) *Given a set $C = \{C_1, ..., C_Q\}$ of subsets of A, corresponding relevant statement sets R_i for each $C_i \in C$, conflict-free preference sets $\overline{R}_i$ for each R_i, and subutility functions u_i 1-consistent with $\overline{R}_i$, a generalized additive utility function u is consistent with a preference r if for all $(b_1, b_2) \models r$, we have*

$$\sum_{i\in C_a(r)} t_i(u_i(b_1) - u_i(b_2)) > \sum_{j\in C_d(r)} t_j(u_j(b_2) - u_j(b_1)). \tag{9.6}$$

Proof. Given in the appendix.

There are two important consequences of this theorem.

Corollary 9.3.1 (Total Agreement) *A subutility utility function u_i for the attributes C_i* must *be 1-consistent with $S \in R_i$ on C_i if $S = p \succ_{CP} q$ and $s(p) \cup s(q) \subseteq C_i$.*

In this case we have exactly one index i such that $S \in R_i$. Thus, either the subutility function is 1-consistent with S and we have $|C_a(r)| = 1$ and $|C_d(r)| = 0$, or the opposite, the subutility function is not 1-consistent with S and we have $|C_a(r)| = 0$ and $|C_d(r)| = 1$. It is easy to see in the second case that inequality 9.6 cannot be satisfied, since $t_i(u_i(b_2) - u_i(b_1))$ is a positive quantity for $i \in C_d(r)$, and this cannot be less than zero. Thus, if a preference's support attributes overlap with only one subutility, then that subutility must be consistent with the preference.

Corollary 9.3.2 (Minimal Agreement) *In an additive decomposition utility function u consistent with M, each statement $S \in M$ must be 1-consistent with some subutility function u_i.*

By assumption, u satisfies Theorem 9.7, so u must satisfy inequality 9.6. If some statement S is not 1-consistent with any subutility function u_i, then we have $|C_a(r)| = 0$ and $|C_d(r)| \geq 1$, and inequality (9.6) will not be satisfied.

Using the above theorem and corollaries, a number of algorithms are possible to produce a consistent utility function. We discuss these in the following subsection.

9.3.4 Choosing Scaling Parameters

Once we have a collection of generalized additively independent sets C, we create subutility functions as described in [MD04]. We can then choose scaling parameters t_i based on which statements disagree with each partial utility function u_i. There are several possible strategies for choosing these parameters. We will first give the most simple scenario, and follow that with the general method.

9.3.5 No Conflicting Preferences

Given a set of qualitative *ceteris paribus*preferences M, a set C of subsets of A, and corresponding relevant statement sets R_i, the sets R_i may or may not have conflicting preferences. The simplest possible scenario is when there are no conflicting preferences on any sets $C_i \in C$, and each u_i is 1-consistent with each preference R_i. If this is the case, consistency of u becomes easy to achieve. Consider that inequality 9.6 becomes:

$$\sum_{i \in C_a(r)} t_i(u_i(b_1) - u_i(b_2)) > 0$$

since the set of inconsistent subutility indices, $C_d(S)$, is empty. By the definition of consistency with S, for $i \in C_a(S)$, $t_i(u_i(b_1) - u_i(b_2))$ is always a positive quantity. Thus, we are free to choose any positive values we wish for t_i when u_i is positively ordered, and negative values when u_i is negatively ordered. For simplicity, choose $t_i = 1$ for positively ordered subutility functions and $t_i = -1$ for negatively ordered subutility functions.

This brief inquiry lets us state the following theorem.

Theorem 9.8 (Simple Scaling Parameters) *Given a set of attribute sets $C = \{C_1, C_2, ..., C_Q\}$ and a set of qualitative* ceteris paribus*preferences M in $\mathcal{L}(A)$, with R_i the set of relevant statements for C_i, and each u_i is 1-consistent with R_i, then*

$$u(\vec{a}) = \sum_{i=1}^{Q} u_i(\vec{a}) \tag{9.7}$$

is an ordinal utility function consistent with M.

Proof. Given in the appendix.

As a corollary, under the same conditions, each of the graphical utility functions can be made consistent with M.

Corollary 9.3.3 (No Conflicting Preferences) *Given a set of attribute sets $C = \{C_1, C_2, ..., C_Q\}$ and a set of qualitative* ceteris paribus*preferences M in $\mathcal{L}(A)$, with R_i the set of relevant statements for C_i and R_i conflict-free for all i, then*

$$u(\vec{a}) = \sum_{i=1}^{Q} u_i(\vec{a}) \tag{9.8}$$

where each u_i is one of the graphical utility functions given in [MD04] over the restricted model graph $G_i(R_i)$, is an ordinal utility function consistent with M.

Proof. By assumption, R_i is conflict-free, so the restricted model graph $G_i(R_i)$ is cycle-free. By Theorem 9.6 each u_i is 1-consistent with each $r \in R_i$. By Theorem 9.8, u is consistent with M. □

9.3.6 Scaling Parameter Assignment by Constraint Satisfaction

When we have a set of attribute sets C of A without a conflict-free ordering, we can construct a constraint satisfaction problem from the set of conflicts and use a general boolean SAT-solver to find a solution. The satisfiability problem will be constructed so that it represents the statement conflicts on the utility independent attribute sets, and a solution to the constraint problem will determine which subutility functions agree and disagree with each statement.

Given qualitative *ceteris paribus*preferences M and a set of attribute sets C of A with corresponding relevant statement sets R_i, we define a boolean satisfiability problem $P(M, C)$ in conjunctive normal form (CNF), a conjunction of disjunctions. Let Y_{ik} be a minimal set of conflicting statements on R_i, and let $Y_i = \{Y_{i1}, Y_{i2}, ..., Y_{iK}\}$ be a minimal and complete set of conflicts for C_i. Let Y be the set of all such Y_{ik}. And let

$$R_c = \bigcup_i \bigcup_k Y_{ik}$$

the set of all statements involved in any conflict. The boolean variable z_{ij} in $P(M, C)$ represents a pair (i, S_j), where $S_j \in M$ and i is an index of $C_i \in C$. The truth value of z_{ij} is interpreted as stating whether or not subutility function u_i agrees with statement $S_j \in R_c$. Let $X_j = \{l \mid (b_1, b_2) \models S_j \implies (b_1 \upharpoonright C_l) \neq (b_2 \upharpoonright C_l)\}$ denote the set of indices of utility independent sets that overlap with S_j.

Conjuncts in $P(M, C)$ are of one of two forms: those representing the subutility functions in a set X_j; and those representing conflicts between statements of a particular cycle Y_{ik}.

Definition 9.1 (Statement-Set Clauses) *Given statements M, a set of attribute sets C of A with corresponding relevant statement sets R_i, R_c of all statements involved in any conflict and X_j the set of indices of utility independent sets overlapping with $S_j \in R_c$, then all clauses of the form*

$$\bigvee_{i \in X_j} z_{ij} \tag{9.9}$$

are the Statement-Set Clauses of $P(M, C)$.

The statement-set clauses of $P(M, C)$ represent the possible subutility functions a statement might agree with. Corollary 9.3.2 states each statement $S \in M$ must be 1-consistent with some subutility function u_i, accordingly in a solution to $P(M, C)$, one of these variables must be true.

Definition 9.2 (Conflict Clauses) *Given preferences M, a set of attribute sets C of A with corresponding relevant statement sets R_i, $Y_i = \{Y_{i1}, Y_{i2}, ..., Y_{iK}\}$ be a minimal and complete set of conflicts for C_i, R_c of all statements involved in any conflict* *and X_j the set of indices of utility independent sets overlapping with $S_j \in R_c$, then all clauses of the form*

$$\bigvee_{j:S_j \in Y_{ik}} \neg z_{ij} \tag{9.10}$$

are the Conflict Clauses of $P(M, C)$.

The conflict clauses of $P(M, C)$ represent a particular conflicts set of statements on a particular utility independent set C_i. At least one of the statements in Y_{ik} must disagree with C_i.

Combining statement-set clauses and conflict clauses, $P(M, C)$ is the conjunction of all such clauses. Thus

$$P(M, C) = \left(\bigwedge_{j=1}^{Q} \left(\bigvee_{i \in X_j} z_{ij} \right) \right) \wedge \left(\bigwedge_{Y_{ik} \in Y} \left(\bigvee_{j:S_j \in Y_{ik}} \neg z_{ij} \right) \right)$$

Once we have computed $P(M, C)$, we can use a solver to arrive at a solution Θ. This solution Θ is an assignment of *true* or *false* to each variable z_{ij} in $P(M, C)$. Note that nothing guarantees that there will be a solution Θ.

Given a solution Θ to $P(M, C)$, we need to translate that back into a construction for the subutility functions over C. Using the definition of the propositional variables z_{ij}, we look at the truth values assigned to z_{ij} in Θ. This shows which subutility functions disagree with which preferences. Let Θ_f be the set of all z_{ij} assigned value *false* in Θ. We define statement sets $\overline{R}_i \subseteq R_i$ for each relevant statement set R_i, as follows. For all $C_i \in C$,

$$\overline{R}_i = R_i \backslash \{S_j \mid z_{ij} \in \Theta_f\} \tag{9.11}$$

Simply, we construct conflict-free relevant statement sets $\overline{R}_i \subseteq R_i$, from the solution to $P(M, C)$, then we can construct u_i 1-consistent with $\overline{R}_i$.

We show that $\overline{R}_i$ constructed according to Θ are conflict-free.

Lemma 9.3.3 (Conflict Clause Solution) *Given a solution Θ to $P(M, C)$, and relevant statement sets R_i for each $C_i \in C$, statement sets $\overline{R}_i \subseteq R_i$,*

$$\overline{R}_i = R_i \backslash \{S_j \mid z_{ij} \in \Theta_f\}$$

are conflict-free.

Proof. We proceed with a proof by contradiction. Choose some C_i. Assume $\overline{R}_i$ has a conflict, $Y_{\overline{R}} \subseteq \overline{R}_i$. Since Y is a complete set of conflicts, $Y_{\overline{R}_i} \in Y$. Thus, there exists a conflict clause $\vee_{j:S_j \in Y_{\overline{R}_i}} \neg z_{ij}$ in $P(M, C)$, so one of these variables $z_{ij} \in \Theta_f$. Call this variable z_{ix}. By definition of $\overline{R}_i$, when $z_{ix} \in \Theta_f$, statement $S_x \notin \overline{R}_i$. This contradicts $S_x \in \overline{R}_i$, so we have arrived at a contradiction and must conclude that $\overline{R}_i$ is conflict-free. □

Theorem 9.9 (Correctness of SAT-Formulation) *Given a set of attribute sets C of A, if $P(M, C)$ has no solution then the preferences M are not consistent with a utility function u of the form $u(\vec{a}) = \sum_i t_i u_i(\vec{a})$.*

Proof. Given in the appendix.

Suppose there is no solution Θ. One method of proceeding is to merge two attribute sets C_i, C_j. We can define $C' = C$, except that C' contains the set C_k, where $C_k = C_i \cup C_j$, and C' does not contain either of C_i and C_j. Using C', we can obtain another satisfiability problem, $P(M, C')$. This new problem may have a solution or it may not. In the case that it doesn't, we may iteratively try merging attribute sets and solving the resulting satisfiability problems. Merging attribute sets is an operation that does not create conflicts where there were none, as we demonstrate in the following theorem.

Theorem 9.10 (Set Merging) *If M is an inconsistent preference set over attributes C_k, then there does not exist any C_i, C_j with $C_i \cup C_j = C_k$ such that M restricted to C_i is consistent and M restricted to C_j is consistent.*

Proof. Given in the appendix.

By the contrapositive, Theorem 9.10 implies that joining two sets C_i, C_j together to form C_k will not result in a set with conflicts if C_i, C_j are conflict-free. Thus, this theorem implies that iteratively merging attribute sets is guaranteed to result in a solvable satisfiability problem when M is consistent. If the preferences M are consistent, by definition the set A is conflict free, then iterative set-merging must end when all attributes have been merged into one set identical with A. How best to choose which attribute sets to merge is a topic we leave for future research.

By merging utility independent sets, the computation of $u(\vec{a})$ becomes less efficient, and we spend computational effort to evaluate several different satisfiability problems. In general, merging attribute sets allows us to represent more complicated preferences, such as utility dependence (including dependencies missed by function *UIDecomposition*). Merging attribute sets is a technique that applies to different representations of *ceteris paribus*preferences. For example, Boutilier, Bacchus, and Brafman [BBB01], mention that merging attribute sets allows their representation to express *ceteris paribus*preferences which would be inexpressible in their representation without merging.

Even in the case that a solution to $P(M, C)$ exists, we are not guaranteed that it leads to a consistent utility function. There is still the matter of setting the scaling parameters t_i. In doing so, we must be careful to satisfy inequality (9.6). Therefore, for each statement $S \in R_i \backslash \overline{R}_i$, we keep a list of linear inequalities, I, that must be satisfied by choosing the appropriate scaling parameters t_i. We discuss this in the following subsection.

9.3.7 Adding Tradeoffs to Qualitative *Ceteris Paribus* Preferences

We are going to construct three lists of linear inequalities, I, I', I'', that must be satisfied by choosing appropriate subutility function parameters t_i. These constraints come from our list of qualitative *ceteris paribus*preference statements M, the list of conditional basket tradeoffs M', and the list of conditional attribute tradeoffs,

M'', respectively. Together these three lists form a system of linear inequalities that represent the preferences in M, M', and M''. It is a simple matter to obtain linear inequalities from M' and M'' if the tradeoff statements result in linear constraints. This, in turn, is the case when the subutility functions are linear, or, by Theorem 7.4, when subutility functions are over discrete attributes and have constant slope.

Let M' and M'' be sets of tradeoff and importance statements, respectively: both conditional basket tradeoffs (CBT) and conditional attribute tradeoff statements (CAT). For each CBT statement $S \in M'$, where $S = d \succ rd'$ (and recall that d and d' are in disjunctive normal form), let X be the set of conjunctive clauses of d, and Y the set of conjunctive clauses of d'. Then for each basket $x \in X$ and each basket $y \in Y$, we have one basket tradeoff $x \succ ry$, or by Theorem 6.3, we have a constraint:

$$\sum_{i=1}^{Q} \sum_{f_j \in s(x) \cap C_i} t_i x_j > r \sum_{k=1}^{Q} \sum_{f_l \in s(y) \cap C_k} t_k y_l. \tag{9.12}$$

Let the set of these constraints for all $S \in M'$ be the set of linear inequalities I'.

Then if we consider all the attribute statements $S' \in M''$, we will obtain additional linear inequalities bounding the tradeoff parameters t_i of the utility function. For each attribute tradeoff statement $S' \in M''$ with $S = D \rhd rD'$, where D, D' are attribute clauses, let X be a set of attributes equivalent to D, and Y be a set of attributes equivalent to D'. Then S' implies the constraint:

$$\sum_{i=1}^{Q} |X \cap C_i| t_i > r \sum_{j=1}^{Q} |Y \cap C_j| t_j. \tag{9.13}$$

And we shall let the set of these constraints for all $S' \in M''$ be the set of linear inequalities I''.

We will discuss conditional statements for conditions other than $\vec{A} \implies S$ in a following section.

9.3.8 Linear Inequalities

We will construct a set of linear inequalities relating the scaling parameters t_i in u from three sources. The first set of linear inequalities is I, which comes from the qualitative *ceteris paribus*preferences statements M. The second set is I', described in the preceding section, and comes from a set of basket tradeoff statements M'. The third set of inequalities is I'', and comes from a set of attribute tradeoff statements M''.

The solution to a satisfiability problem $P(M, C)$ determines how to construct conflict-free preference sets $\overline{R}_i \subseteq R_i$ for each attribute set $C_i \in C$. However, we must then set the scaling parameters, t_i, so that inequality 9.6 is satisfied. We can do so by building a list of constraints on the values that the scaling parameters may assume while still making inequality 9.6 true. These constraints are linear inequalities relating the relative importance of the parameters t_i.

Our list I of inequalities should assure that each pair (b_1, b_2) that satisfies S, for some $S \in M$, is consistent with the total utility function. By inequality 9.6, given statements M, a set of attribute sets C of A with corresponding relevant statement sets R_i, and conflict-free $\overline{R_i} \subseteq R_i$, let $C_a(S)$ be the set of indices for which $S \in \overline{R_i}$, and let $C_d(S)$ be the set of indices for which $S \in (R_i \backslash \overline{R_i})$. Then for each $(b_1, b_2) \models S$

$$\sum_{i \in C_a(S)} t_i u_i(b_1) + \sum_{i \in C_d(S)} t_i u_i(b_1) > \sum_{i \in C_a(S)} t_i u_i(b_2) + \sum_{i \in C_d(S)} t_i u_i(b_2). \tag{9.14}$$

That is, the utility function u must assign higher utility to b_1 than to b_2. We can simplify these inequalities as follows:

$$\sum_{i \in \{C_a(S) \cup C_d(S)\}} t_i(u_i(b_1) - u_i(b_2)) > 0 \tag{9.15}$$

Note that the inequality need not contain terms for $i \notin \{C_d(S) \cup C_a(S)\}$, since, by definition of $(b_1, b_2) \models S$, we have $(b_1 \upharpoonright C_i) = (b_2 \upharpoonright C_i)$ and $u_i(b_1) - u_i(b_2) = 0$. Furthermore, note that many of the linear inequalities are redundant. A simple intuition tells us the important inequalities are the boundary conditions: the constraints that induce maximum or minimum values for the parameters t_i. Consider the set of basket pairs (b_1, b_2) such that $(b_1, b_2) \models S$. Some pairs provide the minimum and maximum values for $u_i(b_1) - u_i(b_2)$ for some $i \in \{C_a(S) \cup C_d(S)\}$. Call $minmax(S, i)$ the set of maximum and minimum values of $u_i(b_1) - u_i(b_2)$ for basket pairs satisfying S. This set has at most two elements and at least one. Let $\mathcal{B}(S)$ be the set of basket pairs as follows:

$$\begin{aligned}
&\mathcal{B}(S): \\
&(b_1, b_2) \models S, \\
&u_i(b_1) - u_i(b_2) \in minmax(S, i) \forall i \in \{C_a(S) \cup C_d(S)\}.
\end{aligned}$$

Each basket pair in the set $\mathcal{B}(S)$ causes $u_i(b_1) - u_i(b_2)$ to achieve either its minimum or its maximum value on every subutility function either agreeing or disagreeing with S. For concreteness, stipulate that each $(b_1, b_2) \in \mathcal{B}(S)$ has $v_j(b_1) = 0$ and $v_j(b_2) = 0$ for $j \notin \{C_a(S) \cup C_d(S)\}$. We can now define the irredundant system of linear inequalities as an inequality of the form given in equation 9.15 for each statement S and each basket pair $(b_1, b_2) \in \mathcal{B}(S)$.

Definition 9.3 (Inequalities) *Given conditional* ceteris paribus*preference statements M, a set of attribute sets C of A with corresponding relevant statement sets R_i, conflict-free $\overline{R_i} \subseteq R_i$, $\overline{R_i} \in \overline{R}$, where $C_a(S)$ is the set of indices for which $S \in \overline{R_i}$, $C_d(S)$ is the set of indices for which $S \in (R_i \backslash \overline{R_i})$, and each subutility function u_i is 1-consistent with all $S \in \overline{R_i}$, then $I(M, C, \overline{R})$ is the set of linear inequalities*

$$\sum_{i \in \{C_a(S) \cup C_d(S)\}} t_i(u_i(b_1) - u_i(b_2)) > 0$$

for all S in some $(R_i \backslash \overline{R_i})$ and for all basket pairs in $\mathcal{B}(S)$.

A solution to $I(M, C, \overline{R})$ is an assignment of values to each t_i. We show that this assignment leads to a utility function that satisfies M.

Lemma 9.3.4 (Sufficiency of $I(M, C, \overline{R})$) *Given $I(M, C, \overline{R})$ and a solution to it providing an assignment of values to $t_1, ..., t_Q$, for all S in some $(R_i \backslash \overline{R_i})$ and for all basket pairs $(b_1, b_2) \models S$,*

$$\sum_{i \in \{C_a(S) \cup C_d(S)\}} t_i(u_i(b_1) - u_i(b_2)) > 0.$$

Proof. Given in the appendix.

Theorem 9.11 (Qualitative Preference Inequalities) *If the system of linear inequalities, $I(M, C, \overline{R})$, has a solution, this solution corresponds to a utility function u consistent with M.*

Proof. Given in the appendix.

Similarly, the systems of inequalities I', I'', also provide consistency of the utility function with the tradeoff and attribute preferences they represent. This, of course, is by definition of those preferences. Thus, the interesting theorem follows from Theorem 9.11.

Theorem 9.12 (General Inequalities) *If the system of linear inequalities, $I(M, C, \overline{R}) \cup I' \cup I''$, has a solution, this solution corresponds to a utility function u consistent with $M \cup M' \cup M''$.*

Proof. This follows directly from Theorem 9.11, since adding more constraints cannot invalidate the previous constraints. The solution to this problem is still consistent with M because the solution to $I(M, C, \overline{R})$ is consistent with M. □

We can solve the system of linear inequalities $I(M, C, \overline{R}) \cup I' \cup I''$ using any linear inequality solver. We note that it is possible to phrase this as a linear programming problem, and use any of a number of popular linear programming techniques to find scaling parameters t_i. We can rewrite the inequalities in $I(M, C, \overline{R})$ in the form

$$\sum_{i=1}^{Q} t_i(u_i(b_1) - u_i(b_2)) \geq \epsilon \tag{9.16}$$

where ϵ is small and positive. Then we can rewrite the inequalities I' in this form,

$$\sum_{i=1}^{Q} \sum_{j: f_j \in (s(x) \cup s(y)) \cap C_i} t_i(x_j - r y_j) > \epsilon,$$

where, again, the notation is that of inequality 9.12. And lastly, we can rewrite the inequalities of I'' as show in inequality 9.13 as

$$\sum_{i=1}^{Q} |(X \cup Y) \cap C_i| t_i - r t_i > \epsilon.$$

Using these altered forms, we can then solve this system of inequalities by optimizing the following linear program. The task of the linear program is to minimize a new variable, t_0, subject to the following constraints:

$$t_0 + \sum_{i=1}^{Q} t_i(u_i(b_1) - u_i(b_2)) \geq \epsilon$$

for each (b_1, b_2) as described above,

$$t_0 + \sum_{i=1}^{Q} \sum_{j:f_j \in (s(x) \cup s(y)) \cap C_i} t_i(x_j - ry_j) > \epsilon$$

for each inequality in I',

$$t_0 + \sum_{i=1}^{Q} \sum_{j:f_j \in (X \cup Y) \cap C_i} t_i - rt_i > \epsilon$$

for each inequality in I'', and the final additional constraint that $t_0 \geq 0$ [Chv83]. We must be careful that ϵ is sufficiently smaller than 1, or we may not find solutions that we could otherwise discover. Given ϵ small enough, this linear program has the property that it always has a solution, by making t_0 very large. However, only when the optimal value of t_0 is 0, is there a solution to the original system of linear inequalities.

We can bound the total number of linear inequalities in $I(M, C, \overline{R})$ as follows. Given that $S \in (R_i \backslash \overline{R}_i)$ for some i, it contributes at most

$$2^{|C_a(S) \cup C_d(S)|} \tag{9.17}$$

inequalities to the set of inequalities. This is a simple combinatoric argument. S contributes exactly one inequality for each combination of minimum and maximum value on each set C_i. For each set a statement can have both a minimum and a maximum (they may not be distinct). This suffices to give us the bound stated in condition 9.17.

The number of inequalities in I' is determined by the sizes of the clauses in the tradeoff preferences. First consider a basket tradeoff preference $S = \vec{A} \implies d \succ_{CP} rd'$. The clauses d, d' are given in disjunctive normal form. For each $x \in d$ and $y \in d'$, we treat $x \succ_{CP} ry$ as a separate, simplified, preference. Thus the number of inequalities contributed per tradeoff preference S is $|d| * |d'|$.

Attribute tradeoff statements are simpler and contribute one inequality per preference to I''.

We have mentioned that $I(M, C, \overline{R}) \cup I' \cup I''$ might have no solution. We discuss this situation in the following subsection.

9.3.9 Piecewise Linear Utility Functions

When preferences are conditional, and hold at different regions in the outcome space $\vec{A}$, these different preferences imply different constraints on the utility function in these different regions. In general these constraints are not simultaneously satisfiable, and this necessitates different utility functions for different regions of the space. In this way, the utility function for one region can satisfy the constraints required of the utility function in that region, and a utility function for another region can satisfy the constraints for that region. The utility function for the whole attribute space $\vec{A}$ is then a collection of different utility functions for different subsets of the attribute space. Since each of these utility functions are linear, the whole becomes a *piecewise linear utility function.*

Definition 9.4 (Piecewise Utility Function) *U is a* piecewise utility function *if U is a set of pairs (V, u_V) where V is a compound value proposition and $u : \vec{A} \rightarrow \Re$ is a utility function.*

U assigns the value $u_V(\vec{x})$ to $\vec{x}$ when $\vec{x} \in [V]$. We write $U(\vec{x})$ for the value U assigns to $\vec{x}$. In this way, a piecewise utility function is like a *switch* or *case* statement in a programming language; it selects which of several utility functions to use based on the input. With this definition in hand, we proceed with the discussion.

When the utility functions are linear in each of the attributes, different constraints on the utility function are the result of conditional preferences. This is straightforward; different preferences can be conditioned on different regions of the space, using the conditional preferences provided in the language $\mathcal{L}(A)_1$ and $\mathcal{L}(A)_2$. In this way, different preferences lead to different constraints on the utility function.

The conditional tradeoffs expressed in $\mathcal{L}(A)_1$ and $\mathcal{L}(A)_2$ are binary conditions. In some region of the attribute space, the preference holds, and in the remainder of the attribute space, the preference does not apply. Thus, for each conditional preference, we can divide the attribute space into these two pieces. Given k conditional statements, each with independent conditions, we are left with as many as 2^k separate divisions of the attribute space with different preferences holding in each division.

Given k conditional tradeoffs, we can define the 2^k subsets of the attribute space as follows. For a set $\{M' \cup M''\}$ of k conditional preference statements, each region of the attribute space is defined by the intersection of a unique subset of the k conditions. Let W be the set of condition clauses corresponding to M', then each $w \in W$ is a separate compound value clause. Any subset $V \subseteq W$ holds on a region of the attribute space defined by $\bigwedge_{w \in V}$. For example, if there are k conditional preference statements $S_1, S_2, ..., S_k$, each with corresponding condition clause $W_1, W_2, ..., W_k$, then there is a subset of the attribute space defined by $W_1 \wedge W_4 \wedge W_5$ where statements S_1, S_4, and S_5 hold, and so on for each subset of conditions.

For each subset V of W, the set of tradeoff preferences that hold over the corresponding space is just those that correspond to the conditions. We state this in the following theorem.

Theorem 9.13 (Space Conditions) *Given a set of conditional preferences* $\{M' \cup M''\}$*, with corresponding conditions* W*, each subset* $V \subseteq W$ *defines a region of the attribute space* $\bigwedge_{w \in V} w$ *where preferences corresponding to* V *in* M' *hold.*

Proof. This theorem follows directly from the definition of conditional preference. □

Note that if condition $\bigwedge_{w \in V} w$ is unsatisfiable, then V describes an empty section of the attribute space, and this section does not require further consideration.

This theorem defines the regions of the space where different constraints hold. However, just because these regions have different constraints, it does not mean that the constraints are mutually exclusive or unsatisfiable. Given a set of conditions W and two subsets, $V \subset W, V' \subset W$, if the utility function constraints holding over $V \cup V'$ are satisfiable by some utility function u', then this utility function can be used for $V \cup V'$.

In the presence of conditional tradeoff preferences, we proceed as follows. For a set of conditional and unconditional tradeoff and attribute preferences $\{M' \cup M''\}$, consider the set W of conditions on those preferences. For each subset V of W, let J be the set of preferences from $\{M' \cup M''\}$ conditioned by V. Then for preferences $J \cup M$, we can construct sets of linear inequalities as discussed in the preceding section; the solution to this set of linear inequalities gives us a utility function. This utility function, in turn, is the utility function for the subset of the attribute space indicated by $\bigwedge_{w \in V} w$. In this way, we construct separate utility functions for different sections of the attribute space.

The methods of dealing with different conditions on tradeoffs here can be computationally difficult. Is is possible that borrowing techniques from constraint satisfaction literature would be efficacious here, but we leave such speculation to future work.

9.3.10 A Detailed Example

Let us consider an example and how it can fit into the frameworks mentioned above. Suppose we are going out to eat in Boston, and need to pick a restaurant. We consider the food, the wine, the atmosphere, the time to get there, and the time spent waiting once at the restaurant. Usually, in Boston, restaurants are crowded, and since we do not have reservations expedience can be a serious concern. Let $\vec{A} = \langle m, w, a, tt, wt \rangle$ for *meal, wine, atmosphere, travel time*, and *wait time*. Then let *meal* have two values: b_1 =*meat*, b_2 =*fish*; *wine* have two values: w_1 =*red*, w_2 =*white*; and *atmosphere* have three values: a_1 =*bland*, a_2 =*gaudy*, a_3 =*quiet*. *Travel time* and *wait time* will be measured in minutes. We now state some simple *ceteris paribus* preferences:

	var	*Preferences*
p_1	m	*fish* $\succ$ *meat*
p_2	w	*fish* $\wedge$ *white* $\succ$ *fish* $\wedge$ *red*
p_3	w	*meat* $\wedge$ *red* $\succ$ *meat* $\wedge$ *white*
p_4	a	*quiet* $\succ$ *gaudy*
p_5	a	*gaudy* $\succ$ *bland*

These preferences mean that we prefer fish to meat. Preferences p_2 and p_3 mean that our preference for wine depends on the main course. The remaining two preferences establish an order over the possible restaurant atmospheres.

Travel time and *wait time* are numeric variables where less is better. We state tradeoffs about these variables: $wt \rhd 1.5tt$, which indicates that is roughly 50% more annoying to wait at the restaurant than to travel to it. These preferences have laid the groundwork for a tradeoff between groups of variables: $\{m, w, a\} \rhd 10\{tt, wt\}$. This implies the following condition on the partial derivatives of the utility function:

$$\frac{\partial u}{\partial m}(\vec{x}) + \frac{\partial u}{\partial w}(\vec{x}) + \frac{\partial u}{\partial a}(\vec{x}) \geq 10\left(\frac{\partial u}{\partial tt}(\vec{x}) + \frac{\partial u}{\partial wt}(\vec{x})\right). \tag{9.18}$$

We can now construct subutility functions for each of the attributes, or, in this case, for each utility independent set of attributes. Here attribute w is utility dependent on attribute m, so following the system of section 9.2.1, we generate one subutility function for $\{m, w\}$, one subutility function for a, one for tt, and one for wt. For the qualitative attributes, we can specify their subutility functions simply by assigning numbers to each of the qualitative alternatives of each attribute, and using these assignments as the output of the subutility function for these attributes, respectively. To continue this example, let us assign subutility functions as follows.

Subutility	*Value*	*Subutility*	*Value*
$u_{\{m,w\}}$(*fish, white*)	3	u_a(*quiet*)	3
$u_{\{m,w\}}$(*fish, red*)	2	u_a(*gaudy*)	2
$u_{\{m,w\}}$(*meat, red*)	2	u_a(*bland*)	1
$u_{\{m,w\}}$(*meat, white*)	1		

For numeric attributes wt and tt, we can choose a simple linear subutility function. We take $u_{wt} = -wt$ and $u_{tt} = -tt$.

The subutility functions are now known, and the form of the utility function (additive) is known, that is, the utility function is of the form: $u(\vec{a}) = \sum_i t_i u_i(\vec{a})$. But before we can use the inequalities involving the partial derivatives of the utility function, we must assign value functions that take the discrete domains to numbers. We proceed in the most straightforward way, and assign values as follows:

Value Function	*Value*	*Value Function*	*Value*
ρ_m(*fish*)	2	ρ_a(*quiet*)	3
ρ_m(*meat*)	1	ρ_a(*gaudy*)	2
ρ_w(*white*)	2	ρ_a(*bland*)	1
ρ_w(*red*)	1		

Theorem 7.4 lets us use the slope of a linear subutility function as the partial derivative of that subutility function with respect to the value of our ordinal variables, and therefore we can compute the partial derivatives of the utility function and simplify inequality 9.18. We must consider that we have different partial derivatives at different vectors in $\vec{A}$. In particular, when we evaluate the partials of u with respect to m and to w, we let x be in the domain of m, and y be in the domain of w. In these cases we have

$$\begin{aligned} \frac{\partial u}{\partial w}(x,y) &= t_{\{m,w\}}(u_{\{m,w\}}(x, \mathit{white}) - u_{\{m,w\}}(x, \mathit{red}))/(\rho(\mathit{white}) - \rho(\mathit{red})) \\ \frac{\partial u}{\partial m}(x,y) &= t_{\{m,w\}}(u_{\{m,w\}}(\mathit{fish}, y) - u_{\{m,w\}}(\mathit{meat}, y))/(\rho(\mathit{fish}) - \rho(\mathit{meat})) \end{aligned}$$

Note that the other partial derivatives are straightforward. Thus, using the above, when we fix $m = \mathit{fish}$ when computing $\frac{\partial u}{\partial w}(w)$ and $w = \mathit{white}$ when computing $\frac{\partial u}{\partial m}(m)$ we have

$$|2t_{\{m,w\}}| + |t_{\{m,w\}}| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

Similarly if we fix $m = \mathit{meat}$ and $w = \mathit{white}$ then

$$|2t_{\{m,w\}}| + |-t_{\{m,w\}}| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|),$$

and $m = \mathit{fish}$ with $w = \mathit{red}$ gives

$$0 + |t_{\{m,w\}}| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

Finally fixing $m = \mathit{meat}$ and $w = \mathit{red}$ gives this constraint

$$0 + |-t_{\{m,w\}}| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

Some of these constraints are identical because of the absolute value functions[1], so we can collect cases into two, and have

$$\begin{array}{ll} 3t_{\{m,w\}} + t_a \geq 10(t_{tt} + t_{wt}) & w = \mathit{white} \\ t_{\{m,w\}} + t_a \geq 10(t_{tt} + t_{wt}) & w = \mathit{red}. \end{array}$$

These constraints can then be collected with all other constraints on the parameters of the utility function. Considering the tradeoff preferences above, we now have the following constraints on the parameters of the utility function:

	Constraint	
c_1	$3t_{\{m,w\}} + t_a \geq 10(t_{tt} + t_{wt})$	$w = \mathit{white}$
c_2	$t_{\{m,w\}} + t_a \geq 10(t_{tt} + t_{wt})$	$w = \mathit{red}$
c_3	$t_{wt} \geq 1.5t_{tt}$	

These systems of linear inequalities can be solved for the different cases, in principle resulting in piece-wise linear utility functions. In this case, since constraint c_1 follows from constraint c_2, there is no need to have different functional forms of the utility function based on different values of the w variable. Therefore, a solution for this construction is $t_{m,w} = 15, t_a = 10, t_{wt} = 1.5$, and $t_{tt} = 1$.

Thus a utility function for this example is

$$u(\vec{x}) = 15u_{m,w}(\vec{x}) + 10u_a(\vec{x}) + 1.5u_{wt}(\vec{x}) + u_{tt}(\vec{x}).$$

[1]When preferences for X exactly reverse given different values of Y, this is termed *generalized preference independence* of X of Y by Keeney and Raifa [KR76].

9.3.11 Complexity

The running time of the SAT algorithm and the Linear Inequality solver steps can be prohibitive. We consider here the time taken for both steps, for some different preference structures. We will discuss the following cases: when each attribute $a_i \in A$ is utility independent of every other attribute, when none are utility independent of any other attribute, when each utility independent attribute set is of size equal to the log of the number of attributes in A. There are two different complexity questions to ask. One is the time and space required to construct u. The other is the time and space required to evaluate $u(b)$ on a particular basket b.

In the following, we assume that $|A| = N$ and that the number of utility-independent sets in the partition C is k. We assume there are $|I|$ inequalities from tradeoff preference statements T.

The time required for the SAT solver depends on both the number of clauses and the number of boolean variables appearing in the SAT problem. The number of boolean variables is bounded by $|M| * k$, and the number of clauses depends on the number of conflicts among statements, upon which we do not have a good bound. The SAT clauses are of two forms. The first type ensures that every statement agrees with some utility independent set. Thus, these clauses have one variable per utility independent set C_i such that $(b_1, b_2) \models S \Rightarrow (b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)$. We upper-bound this quantity by k. The second type ensures at least one preference statement of a conflict is ignored. These clauses have a number of boolean variables equal to the number of statements involved in the conflict. The number of statements in each conflict is bounded by $|M|$ and by $m_i^{|C_i|}$, where $m_i = \max_{f \in C_i} |D_f|$, since there can only be at most $m_i^{|C_i|}$ different qualitative *ceteris paribus*preference statements per utility-independent set. In pathological cases, the number of conflicts could be as high as $|M|!$, if every statement conflicts with every other set of statements. However, it is difficult to imagine such scenarios. Particularly, we assume the preferences M are consistent, and will therefore have a low number of conflicts. Clearly, we have no good bound on the number of conflicts. However, since there is no known polynomial-time algorithm for the satisfiability of our CNF formula, the SAT portion of the algorithm will have worst-case exponential time cost unless we can bound the number of conflicts and statements by $O(\log(|A|))$. Simple counterexamples exist that have $O(|A|)$ conflicts. Thus, the efficiency of our algorithm depends on heuristic methods to both 1) reduce the number of conflicts and 2) quickly solve SAT problems. In the first case, we speculate that careful translation of the domain into attributes can reduce the number of conflicts. In the second case, our implementation suggests that fast randomized SAT-solvers (*e.g.,* WalkSAT [SLM92]) quickly solve our constraint problems.

We have already shown that the number of linear inequalities contributed by a qualitative *ceteris paribus*preference statement is less than $2^{|C_a(S) \cup C_d(S)|}$. More precisely, for each statement, there is one inequality for each combination of the maximum and minimum of $u_i(b_1) - u_i(b_2)$ for each subutility function. For the special case where i is such that $|C_i \backslash (s(p) \cup s(q))| = 0$, for $S = p \succ_{CP} q$, then i contributes exactly one term, since the minimum and maximum are the same. Thus each qualitative *ceteris* *paribus*preference statement contributes

$$\prod_{i \in C_a(S) \cup C_d(S)} \left\{ \begin{array}{lll} 2 & if & |C_i \backslash (s(p) \cup s(q))| \geq 1 \\ 1 & if & |C_i \backslash (s(p) \cup s(q))| = 0 \end{array} \right\}$$

linear inequalities.

First we consider two extreme cases. Suppose that every attribute is found to be utility dependent upon every other attribute, so that the partition C is $C_1 = \{a_1, a_2, ..., a_N\}$, and $k = 1$. Then there can be no conflicts, since every qualitative preference statement must agree with u_1. Thus, there are no SAT clauses and no linear inequalities needed. However, the evaluation time of $u(\vec{a})$ is the time taken to traverse a model graph of all N attributes. We argued in [MD04] that this is worst-case exponential in N.

On the other hand, suppose that every attribute is found to be utility independent of every other attribute, and we have $k = N$. The number of inequalities per qualitative *ceteris paribus*preference statement is exactly one, we have the total number of inequalities equal to $|M|$. The evaluation time of $u(\vec{a})$ is order N, since we evaluate N graphs of size D_i.

Now suppose we have a more balanced utility decomposition. Suppose each utility-independent set is of size $\log N$. Then $k = N/(\log N)$. The number of inequalities per qualitative preference statement S is less than $2^{|C_a(S) \cup C_d(S)|}$, and since $|C_a(S) \cup C_d(S)| \leq N/\log N$, this is less than $2^{N/\log N}$. Thus the total number of inequalities is less than $|M| * 2^{N/\log N}$. The evaluation of $u(\vec{a})$ requires traversing k graphs, of size exponential in their domains, or each of size less than $|m_i|^{\log N}$, where m_i is as before, the max domain size of an attribute in attribute set C_i. Thus evaluating $u(\vec{a})$ is order $|m_i|N^2/(\log N)$.

If the size of the utility-independent sets gets larger than $\log N$, then we will not have polynomial time evaluation of $u(\vec{a})$. In addition, the number of possible linear inequalities goes up exponentially with the number of utility-independent sets each preference overlaps with.

In conclusion, given a favorable decomposition of the attribute space into utility independent sets, and few conflicts, we can create a utility function in time $O(|I| + \max_S(2^{|C_a(S) \cup C_d(S)|}) + |M|^2 + \max_i 2^{|C_i|})$. The first term is for the number of inequalities, the second term is for the utility independence computation, and the third term is for actually constructing the subutility functions u_i. We can then evaluate $u(\vec{a})$ in time $O(\max_i 2^{|C_i|})$.

9.3.12 Summary

In this chapter we have presented a number of methods of assigning scaling parameters to the utility function $u(m) = \sum_i t_i u_i(m)$ and choosing which subutility functions should be 1-consistent with which statements. Some methods are faster than others. However, the methods presented here are not complete - they may fail to produce a utility function from *ceteris paribus*preferences M. When the methods fail, it is always possible to join partition elements $S_i, S_j \in S$ together to perform constraint satisfaction and linear programming on a smaller problem.

9.4 A Complete Method

In the preceding, we have described several parts of an algorithm for computing with qualitative *ceteris paribus*preference statements, conditional tradeoff preferences, and conditional attribute tradeoffs. This has given us enough tools to accomplish our goal: generating a utility function consistent with various types of input preferences. The algorithm we present takes sets of *ceteris paribus*, tradeoff, and attribute importance preferences M, M', M'', and computes a piecewise utility function U with $U \in [M]$ and for a skeptical interpretation J' of conditional tradeoffs $M' \cup M''$, $U \in [J']$. In the following, we reiterate how these techniques can be woven together into a complete algorithm.

The algorithm takes as input a set M of qualitative *ceteris paribus*preference statements in the language $\mathcal{L}(A)$, a set of tradeoff statements M' in $\mathcal{L}(A)_1$, a set of attribute statements M'' in $\mathcal{L}(A)_2$, and a type of graphical utility function g (from [MD04]). The algorithm outputs a piecewise linear utility function U, a set of pairs (V, u_V) such that each utility functions u_V is consistent with the input preferences at a different region V of the attribute space $\vec{A}$.

The steps of the algorithm are as follows:

1. Initialize U to the empty set.

2. Compute the set of relevant attributes A, which is the support of $M \cup M' \cup M''$.

3. Compute a partition $C' = \{C'_1, C'_2, ..., C'_Q\}$ of A into utility-independent attribute sets, and compute $D = \{D_1, D_2, ..., D_Q\}$ such that C'_i is utility dependent on D_i and utility independent of $A \backslash D_i$, using function *IUDecomposition*, as discussed in section 9.2.1. Let $C = \{C_1, C_2, ..., C_Q\}$ be such that $C_i = (C'_i \cup D_i)$.

4. Construct relevant statement sets R_i for each C_i. R_i is defined to be the set of statements $S \in M$ such that $(b_1, b_2) \models r \Rightarrow (b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)$.

5. Construct the restricted model graph $G_i(R_i)$ for each set of statements R_i.

6. Compute a minimal and complete set Y_i of cycles Y_{ik} for each graph $G_i(R_i)$ such that each cycle Y_{ik} is a set of statements from R_i.

7. Construct the satisfiability problem $P(M, C)$ from all cycles Y_{ik}.

8. Find a solution Θ to $P(M, C)$.

9. Choose conflict-free preference sets $\overline{R_i} \subseteq R_i$ sets using solution Θ of $P(M, C)$, as given in equation 9.2.

10. Construct cycle-free restricted model graphs $G'_i(\overline{R_i})$

11. Define each subutility function u_i to be the graphical subutility function of type g based on $G'_i(\overline{R_i})$.

12. Use definition 9.3 to construct $I(M, C, \overline{R})$, a system of linear inequalities relating the parameters t_i.

13. Consider each conditional preference in M' and in M'', and let W be the set of conditions upon these preferences. For each set $V \subseteq W$, with non-empty $[V]$, let J be the set of preferences conditioned by V.

    (a) Let $J' = \textit{Skep-Iterp}(J, \vec{x})$, for $\vec{x} \in [V]$, to obtain a skeptical set of preferences J' for J.

    (b) Convert tradeoff statements in J' to a set of linear inequalities I' by using the definition of tradeoff statements and inequality (9.12).

    (c) Convert attribute importance statements in J' to a set of linear inequalities I'' by using the definition of attribute tradeoff statements and inequality (9.13).

    (d) Solve $I(M, C, \overline{R}) \cup I' \cup I''$ for each t_i using linear programming.

        i. If $I(M, C, \overline{R}) \cup I' \cup I''$ has a solution, pick a solution, and use the solution's values for u_i and t_i to construct a utility function $u_V(\vec{a}) = \sum_i t_i u_i(\vec{a})$, and updating $U \leftarrow U \cup \{V, u_V\}$

        ii. If $I(M, C, \overline{R}) \cup I' \cup I''$ has no solution, and $I' \cup I''$ has no solution, then the input tradeoffs are inconsistent; output the empty set.

        iii. If $I(M, C, \overline{R}) \cup I' \cup I''$ has no solution, and $I' \cup I''$ has a solution, then output the empty set.

14. Output U.

Some remarks must be said about the failure conditions of this algorithm. First of all, the algorithm may fail because the steps concerning merely the qualitative *ceteris paribus*preferences can fail; these are heuristic methods. As we discuss in [MD04], consistent *ceteris paribus*preferences can always be represented by a trivial utility function; one that orders each outcome according to the preorder implied by the preferences, but this gains none of the advantages of a generalized additive decomposition utility function.

Secondly, a set of tradeoff preferences cannot be considered consistent without knowledge of the partial derivatives of the utility function. The partial derivatives of the utility function, in this case, are determined by the generalized additive decomposition of the attribute space. Thus we cannot know with certainty before the algorithm determines the additive decomposition of the attribute space if the tradeoff preferences are consistent or not.

With these shortcomings in mind, we must consider this algorithm heuristic. There is always the possibility of conflicting preferences leading to no solution. However, when this algorithm finds a solution, it is guaranteed to represent the input preferences faithfully. This algorithm, therefore, fulfills its main purpose: it illustrates that tradeoff preferences can in principle be combined with qualitative *ceteris paribus*preferences of the type presented in chapter 5. Indeed, we show in the next chapter that tradeoff preferences can be combined with another representation of qualitative *ceteris paribus*preferences.

The soundness of this algorithm can be proven by reference to the preceding theorems of this thesis. We present this here.

Theorem 9.14 (Soundness) *Given a set of* ceteris paribus*preferences M, a set of tradeoff preferences M', and a set of attribute tradeoffs M'', if the above algorithm produces a piecewise linear utility function U, then $U \in [M \cup M' \cup M'']$.*

Proof. By Theorem 9.1, the partition C is a partition of A. Lemma 9.3.2 implies that a minimal and complete set of conflicts Y_i for preference set R_i can be computed by performing cycle-detection on the restricted model graph $G_i(R_i)$. By definitions 9.1 and 9.2, C, R, Y, are enough to construct a satisfiability problem $P(M, C)$. By Lemma 9.3.3, the solution Θ to $P(M, C)$ allows choosing of $\overline{R}_i$ conflict-free. By Lemma 9.3.2, each restricted model graph $G_i(\overline{R}_i)$ is cycle-free. By theorem 8.1, *Skep-Interp* produces a skeptical interpretation of V. It is then possible to build and solve a set of linear inequalities $I(M, C, \overline{R}) \cup I' \cup I''$, as given in definition 9.3. If $I(M, C, \overline{R}) \cup I' \cup I''$ has a solution, then this solution provides values for each t_i. By Theorem 9.12 $u_V(m) = \sum_i t_i u_i(m)$ is a utility function consistent with $M \cup V$. By definition of piecewise linear utility functions, U is such a function, and $U \in [M \cup M' \cup M'']$. □

Chapter 10

Quantitative Tradeoffs and CP-nets

In general, the tradeoffs and importance preference statements described in this thesis generate linear constraints on the parameters of additive utility functions. These constraints can be easily integrated with any preference or utility estimation system that uses linear inequalities to constrain the parameters of possible utility functions. And since linear models of utilities are so common in practice, the system we have proposed should be widely applicable. In the previous chapter we showed how to combine tradeoff preferences with the method of [MD04]. In the present chapter we show how to combine the linear inequalities generated from our preference tradeoff statements with the CP-nets system. We stress that integration with these two systems are merely representative of other possible integrations.

Methods proposed by Brafman, Domshlak, and Kogan [BDK04] take CP-nets [BBHP99] and TCP-nets [BD02] and generate a utility function consistent with the order implied by the CP-net or TCP-net. These methods use qualitative *ceteris paribus* preference as their input, and output a generalized additive-independent ordinal utility function. When we consider the system we have presented in preceding chapters alongside the systems based on CP-nets, we find there are differences of expressiveness. CP-nets place restrictions on the form of the preference statements, and make independence relationships explicit; the methodology we have presented allows arbitrary qualitative *ceteris paribus* preferences and infers independence from the statements. The restrictions on CP-nets allow strong tractability results, in [BDK04] acyclic TCP-nets always allow efficient utility function construction. Such differences mean that each qualitative preference methodology may be appropriate in different situations.

We now demonstrate that the various quantitative tradeoffs we have developed in the preceding chapters fit easily together with CP-nets.

10.1 Adding Quantitative Tradeoffs to CP-nets

To add quantitative tradeoffs to CP-nets [1] we require two things of the utility function; one, that it should be a generalized additive utility function and two, that it should have linear subutility functions.

If we compile a CP-net into a utility function, basing the methods on [BDK04], then we are committing to using a generalized additive utility function. We can force the subutility functions (termed "factors" in that source) of this utility function to be linear in their input by adding additional inequalities to the system of linear equations that generates the utility function. These conditions assure that our tradeoff statements in $\mathcal{L}(A)_1$ or in $\mathcal{L}(A)_2$ can be easily added to the CP-net.

The system of linear inequalities constructed by Brafman *et al.* has one variable per input per subutility function, so we can add additional inequalities asserting, for a variable X, that $kx_1 \leq 2kx_2 \leq 3kx_3, ...$, together with $kx_1 \geq 2kx_2 \geq 3kx_3, ...$, and $k > 0$, assuring that the output of the subutility function for X is linear in X. The proper ordering among values of X can be found by considering the CP-Family of X ([BDK04], section 3), and computing a different linear program for each possible ordering consistent with the CP-Family. This is a locally-exponential addition to the complexity of utility function construction, so the problem remains in P when the exponents are bounded by a constant.

After assuring the subutility functions are linear in their input, it is simple to solve an additional system of linear inequalities which constrain the tradeoff ratios between subutility functions. This new problem has one variable for each subutility function, representing the weight given to it in the generalized additive utility function, and one or more inequalities for each tradeoff statement $S \in \mathcal{L}(A)_1$ or in $\mathcal{L}(A)_2$. Each tradeoff statement results in linear constraints on the tradeoff parameters of the utility function, but may result in different constraints over different areas of the domain of the utility function. This is the case when the preferences over one variable, and thus partial derivatives with respect to that variable, switch with the values assumed by a different variable. Such is the normal case of utility dependence between variables. In these cases, the utility function will be a piecewise linear one, having different functional forms for different parts of its domain.

We demonstrate these techniques in a detailed example in the next section.

10.2 A CP-net Example

In the previous chapter we considered an example involving choosing a restaurant in Boston. We will work through the same example here, again, but this time in a CP-net framework. This illustrates the differences between the CP-net formalism and the methods presented earlier in this thesis.

[1] TCP-nets is a expansion to CP-net semantics that adds qualitative tradeoffs [BD02]. We do not add quantitative tradeoffs to TCP-nets because the semantics of the two tradeoff systems are incompatible.

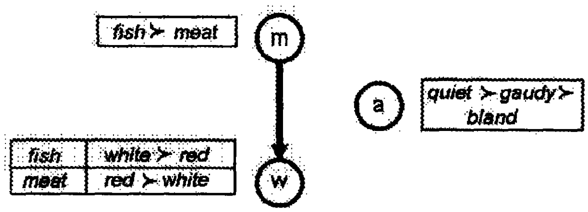

Figure 10-1: CP-net for preferences p_1-p_5

We again choose $\vec{A} = \langle m, w, a, tt, wt \rangle$ for *meal, wine, atmosphere, travel time,* and *wait time*, just as in the previous example.

The simple *ceteris paribus* preferences we used before,

	var	*Preferences*
p_1	m	*fish* $\succ$ *meat*
p_2	w	*fish* $\wedge$ *white* $\succ$ *fish* $\wedge$ *red*
p_3	w	*meat* $\wedge$ *red* $\succ$ *meat* $\wedge$ *white*
p_4	a	*quiet* $\succ$ *gaudy*
p_5	a	*gaudy* $\succ$ *bland*

can be used to construct a CP-net.

In a CP-net for these preferences, we have to consider which variables are utility independent. In this case only w depends on m so we draw the CP-net as shown in figure 1.

Recall that *travel time* and *wait time* are numeric variables, and that we state these two attribute tradeoff preferences over A:

$wt \rhd 1.5tt$
$\{m, w, a\} \rhd 10\{tt, wt\}$

And we have again these conditions on the partial derivatives of the utility function (from inequality (9.18)).

$$\frac{\partial u}{\partial m}(\vec{x}) + \frac{\partial u}{\partial w}(\vec{x}) + \frac{\partial u}{\partial a}(\vec{x}) \geq 10 \left(\frac{\partial u}{\partial tt}(\vec{x}) + \frac{\partial u}{\partial wt}(\vec{x}) \right).$$

These constraints are an addendum to the CP-net framework: we will keep them aside for now.

To compute a utility function for the CP-net we must solve a system of linear inequalities, of the form of inequality 1 in [BDK04]. In this case, it results in the following linear inequalities:

	Preferences
e_1	$u_a(quiet) > u_a(gaudy)$
e_2	$u_a(gaudy) > u_a(bland)$
e_3	$u_w(fish,white) > u_w(fish,red)$
e_4	$u_w(meat,red) > u_w(meat,white)$
e_5	$u_m(fish) + u_w(fish,white) >$ $u_m(meat) + u_w(meat,white)$
e_6	$u_m(fish) + u_w(fish,red) >$ $u_m(meat) + u_w(meat,red)$

We can then add linearizing inequalities to the system, forcing $3k_a u_a(quiet) \geq 2k_a u_a(gaudy) \geq k_a u_a(bland)$ and $3k_a u_a(quiet) \leq 2k_a u_a(gaudy) \leq k_a u_a(bland)$, using a new variable k_a. We make similar inequalities for u_w and u_m. We require these additional inequalities to force the subutility functions for each attribute in the CP-net to be linear; this simplifies our methodology. A solution is as follows:

Subutility	Value	Subutility	Value
$u_m(fish)$	2	$u_w(fish,\ white)$	4
$u_m(meat)$	1	$u_w(fish,\ red)$	3
$u_a(quiet)$	3	$u_w(meat,\ red)$	2
$u_a(gaudy)$	2	$u_w(meat,\ white)$	1
$u_a(bland)$	1	k_a	1
k_m	1	k_w	1

For attributes wt and tt, we use these linear subutility functions : $u_{wt} = -wt$ and $u_{tt} = -tt$.

Recall that we assign value functions that take the discrete domains to numbers, as follows:

Value Function	Value	Value Function	Value
$\rho_m(fish)$	2	$\rho_a(quiet)$	3
$\rho_m(meat)$	1	$\rho_a(gaudy)$	2
$\rho_w(white)$	2	$\rho_a(bland)$	1
$\rho_w(red)$	1		

Theorem 7.4 lets us use the slope of a linear subutility function as the partial derivative of that subutility function with respect to the value of our ordinal variables, and therefore we can compute the partial derivatives of the utility function and simplify inequality 9.18. We have different partial derivatives at different vectors in $\vec{A}$. For example, when we compute the partial derivatives of u with respect to each variable, we must pay special attention to the formulae for the partials of m and of w. For $x \in \{fish,\ meat\}$ and $y \in \{white,\ red\}$, we have

$$\begin{aligned} \frac{\partial u}{\partial w}(x,y) = &\quad t_w(u_w(x, white) - u_w(x, red))/(\rho(white) - \rho(red))) \\ \frac{\partial u}{\partial m}(x,y) = &\quad t_w(u_w(fish, y) - u_w(meat, y))/(\rho(fish) - \rho(meat)))+ \\ &\quad t_m(u_m(fish) - u_m(meat))/(\rho(fish) - \rho(meat)) \end{aligned}$$

Thus, when we fix $m =$*fish* when computing $\frac{\partial u}{\partial w}(w)$ and $w =$*white* when computing $\frac{\partial u}{\partial m}(m)$ we have

$$|t_w| + |3t_w| + |t_m| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

Similarly if we fix $m =$*meat* and $w =$*white* then

$$|-t_w| + |3t_w| + |t_m| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|),$$

and $m =$*fish* with $w =$*red* gives

$$|t_w| + |-t_w| + |t_m| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

Finally fixing $m =$*meat* and $w =$*red* gives this constraint

$$|-t_w| + |-3t_w| + |t_m| + |t_a| \geq 10(|-t_{tt}| + |-t_{wt}|).$$

As we did with the constraints in the last example, we can again collect cases into two, and have

$$\begin{array}{ll} 4t_w + t_m + t_a \geq 10(t_{tt} + t_{wt}) & w = white \\ 2t_w + t_m + t_a \geq 10(t_{tt} + t_{wt}) & w = red. \end{array}$$

These constraints can then be collected with all other constraints on the parameters of the utility function. We then have the following constraints on the parameters of the utility function:

	Constraint	
c_1	$4t_w + t_m + t_a \geq 10(t_{tt} + t_{wt})$	$w = white$
c_2	$2t_w + t_m + t_a \geq 10(t_{tt} + t_{wt})$	$w = red$
c_3	$t_{wt} \geq 1.5t_{tt}$	

The only remaining step is to solve this system of linear inequalities for the tradeoff parameters t. As in the previous example, constraint c_1 follows from constraint c_2, so there is no need to have different functional forms of the utility function based on different values of the w variable. A solution to the CP-net system of inequalities is $t_m = t_w = t_a = 5$, $t_{wt} = 1.5$, and $t_{tt} = 1$.

Thus a utility function for the CP-net is

$$u(\vec{a}) = 5u_m(\vec{a}) + 5u_a(\vec{a}) + 5u_w(\vec{a}) + 1.5u_{wt}(\vec{a}) + u_{tt}(\vec{a}).$$

Chapter 11

Conclusions

We have presented novel methods for enriching systems of qualitative *ceteris paribus* preferences with quantitative tradeoffs of various types over groups of features. These preference systems can then be compiled into quantitative utility functions using modifications of existing techniques. Our work here has provided an important extension to both the systems of [MD02] and [BBHP99].

The main contribution of this thesis has been the representation of tradeoffs, of various types, as constraints on the partial derivatives of the utility function. We have demonstrated that this general approach to tradeoff semantics is broad enough to cover all of the following:

1. Tradeoffs between particular values of attributes.
2. Importance constraints between sets of attributes.
3. Multiattribute tradeoffs of each preference type considered.
4. Tradeoffs over discrete and continuous attributes.
5. Importance judgements between binary attributes.
6. Degenerate tradeoffs between different values of the same attribute.

To formally underpin the connection between these types of preference tradeoffs, we have provided the following equivalences.

1. Basket tradeoffs reduce to constraints on the economic notion of marginal rates of substitution when the relation is between single attributes.
2. Discrete basket tradeoffs reduce to continuous basket tradeoffs in the case of linear utility functions.
3. Discrete attribute tradeoffs reduce to continuous attribute tradeoffs in the case of linear utility functions.
4. Discrete basket tradeoffs reduce to discrete attribute tradeoffs when the baskets involved are equivalent to the characteristic vectors of the sets of attributes related in the attribute tradeoff.

5. Binary *ceteris paribus*attribute tradeoffs reduce to qualitative *ceteris paribus*tradeoffs.

We then have shown that we can combine all of these tradeoff preferences into a single methodology, together with qualitative *ceteris paribus*preferences, for computing a utility function representing preference statements of all forms. Furthermore, these combination methods can function with however many or few preferences happen to be available, these preferences can be over any attributes, and there need be no explicit utility independence given with the preferences. These are all significant departures from the assumptions underlying traditional decision analysis.

11.1 Directions for Future Work

There are many interesting avenues of research left to pursue concerning the interaction of tradeoff constraints and *ceteris paribus*preference statements.

Our representation of tradeoff statements as constraints on the partial derivatives of the utility function is novel, and as such it raises many new questions.

The basis of our interpretation of attribute importance tradeoffs is the ratio of gradients of the utility function. Even so, our use of the magnitude of the gradient of the utility function to calibrate the relative utilities of two subspaces is a heuristic choice. There could be other measures defined for this purpose, such as some kind of average-case improvement measure, for example, that tries to capture the average case outcome. We leave this issue to future research.

Then, there is the interaction of our constraints on the derivatives over nonlinear subutility functions. Sometimes a tradeoff is over utility dependent attributes, which can lead to a preference reversals or other various discrepancies over different parts of the space, depending on the values of the attribute on which the attribute in question depends. This leads to different types of issues when constructing piecewise linear utility functions. When do preference reversals result in mutually incompatible constraints? Do preference reversals require negative values of the subutility function scaling parameters, t_i? It may be possible to characterize the types of subutility functions that either require piecewise linear solutions, or those that do not.

We can investigate the integration of our partial-derivative based tradeoff preferences with other preference reasoning systems. For example, there might yet be a way to combine our tradeoff statements with TCP-nets. A combination with work on answer-set solvers for preference representation is also a possibility [Bre04]. A combination with a machine-learning system based on an SVM architecture [DJ05] may be straightforward, but the combination of techniques in this case may be counterproductive.

We have given very general definitions of the language of preference; ones with the full expected logical generality. However, for purposes of this thesis, only some of these were given meaning. We argued in passing that some of these statements are unlikely, or awkward, and probably have no simple or agreed-upon meaning. However there is still room to explore in more detail which of the remaining statements can be given useful or pleasing semantics. Further, there are more ambitious levels of statements to include. Some philosophers have recommended that statements *ceteris* *paribus*should be replaced by statements that are true only in the worlds most similar to the existing situation [Han89, Han96]. Could we define useful preferences based on this model? Another larger category of preferences to include is metapreferences. There is always the possibility of stating preferences about when to believe other preferences. A useful starting point for this might be to use a temporal ordering. And a last semantic upgrade would be to consider preferences not "everything else equal" but "everything else *almost* equal."

Appendix A

Proofs

A.1 Proofs of Theorems

Theorem 9.4 (Subutilities) *Given statements M and a mutually utility-independent partition C of A, if each u_i is 1-consistent with S, then*

$$u(\vec{a}) = \sum_{i=1}^{Q} t_i u_i(\vec{a})$$

where $t_i(u_i(b) - u_i(b')) > 0$ when $b \succ_{CP} b'$, is consistent with $S \in M$.

Proof. By definition of 1-consistency of u_i with M, $|u_{C_i}(b_1) - u_{C_i}(b_2)| \geq 1$ when $(b_1, b_2) \models S$ and $(b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)$. Since $t_i(u_i(b_1) - u_i(b_2)) > 0$, we have

$$\sum_{i:(b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)}^{Q} t_i(u_i(b_1) - u_i(b_2)) > 0. \tag{A.1}$$

By definition of a subutility function u_i, if $(b_1 \upharpoonright C_i) = (b_2 \upharpoonright C_i)$ then $u_i(b_1 \upharpoonright C_i) = u_i(b_2 \upharpoonright C_i)$, so we have

$$\sum_{i:(b_1 \upharpoonright C_i) = (b_2 \upharpoonright C_i)}^{Q} t_i(u_i(b_1) - u_i(b_2)) = 0. \tag{A.2}$$

Combining equation A.2 and inequality A.1, we have

$$\sum_{i:(b_1 \upharpoonright C_i) = (b_2 \upharpoonright C_i)}^{Q} t_i(u_i(b_1) - u_i(b_2)) + \sum_{i:(b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)}^{Q} t_i(u_i(b_1) - u_i(b_2)) > 0.$$

The above is equivalent to

$$\sum_{i=1}^{Q} t_i(u_i(b_1) - u_i(b_2)) > 0.$$

So we have

$$\sum_{i=1}^{Q} t_i u_i(b_1) > \sum_{i=1}^{Q} t_i u_i(b_2)$$

and

$$u(b_1) > u(b_2)$$

whenever $(b_1, b_2) \models S$. This is the definition of consistency of u with S. Thus, this completes the proof. $\square$

Lemma 9.3.1 (Cycle-Free Subutility) *Given a set of statements $R \subseteq M$ and a set of features $C_i \subseteq A$ such that the restricted model graph $G_i(R)$ is cycle-free, and $u_i(\vec{a})$ is the minimizing graphical utility function over $G_i(R)$, the subutility function $u_i(\vec{a})$ for C_i is 1-consistent with R.*

Proof. We show that $u_i(b_1) \geq 1 + u_i(b_2)$ whenever $(b_1, b_2) \models S$ and $b_1 \upharpoonright C_i \neq b_2 \upharpoonright C_i$. First let u_i be the minimizing graphical utility function (from [MD04]) defined over $G_i(R)$. Then pick some $r \in R$ and some pair $(b_1, b_2) \models r$ where $(b_1 \upharpoonright C_i) \neq (b_2 \upharpoonright C_i)$. By definition of $G_i(R)$, there exists an edge $e(b_1 \upharpoonright C_i, b_2 \upharpoonright C_i)$ thus, $u_i(b_1) \geq 1 + u_i(b_2)$ because the there exists a path from $(b_1 \upharpoonright C_i)$ that contains $(b_2 \upharpoonright C_i)$, and therefore contains the longest path from $(b_2 \upharpoonright C_i)$, plus at least one node, namely the node $(b_1 \upharpoonright C_i)$. So we have $u_i(b_1) \geq 1 + u_i(b_2)$. Since this holds for all $(b_1, b_2) \models r$ and for all $r \in R$, this proves the lemma. $\square$

Lemma 9.3.2 (Conflict Cycles) *Given a set of strict qualitative* ceteris paribus*preference statements $R \subseteq M$ and a set C of feature sets of A, the statements R conflict if and only if the statements R imply a cycle in any restricted model graph $G_i(R)$ of some set of features $C_i \in C$.*

Proof. If the statements R imply a cycle in $G_i(R)$, then there exists some cycle of nodes b_j in $G_i(R)$. Call this cycle $Y = (b_1, b_2, ..., b_k, b_1)$. By the definition of a model graph, an edge $e(b_j, b_{j+1})$ exists if and only if $b_j \succ_{CP} b_{j+1}$ according to R. Thus, for a subutility function u_i to be consistent with R, the following would have to hold:

$$u_i(b_1) > u_i(b_2) > ... > u_i(b_k) > u_i(b_1)$$

Which is not possible, because $u_i(b_1) > u_i(b_1)$ is impossible. Thus, if R implies a cycle in some $G_i(R)$, then there is no subutility function u_i consistent with R.

In the other direction, we show that if the statements R conflict on C_i, then they imply a cycle in $G_i(R)$. We assume the opposite and work toward a contradiction. Suppose R conflict on C_i but they do not imply a cycle in $G_i(R)$. By the definition of a restricted model graph, $G_i(R)$ is cycle-free. Then we define the subutility function u_i to be the minimizing graphical utility function based on the graph $G_i(R)$. By Lemma 9.3.1, u_i is 1-consistent with R. This is a contradiction, so we have shown that when R conflict on C_i they imply a cycle in $G_i(R)$. This proves the lemma. $\square$

Theorem 9.7 (Preference Consistency) *Given a set $C = \{C_1, ..., C_Q\}$ of subsets of A, corresponding relevant statement sets R_i for each $C_i \in C$, conflict-free statement sets $\overline{R}_i$ for each R_i, and subutility functions u_i 1-consistent with $\overline{R}_i$,* *a generalized additive utility function u is consistent with a statement r if for all $(b_1, b_2) \models r$, we have*

$$\sum_{i \in C_a(r)} t_i(u_i(b_1) - u_i(b_2)) > \sum_{j \in C_d(r)} t_j(u_j(b_2) - u_j(b_1)). \tag{A.3}$$

Proof. If u is consistent with r then $u(b_1) > u(b_2)$ whenever $b_1 \succ_{CP} b_2$ according to r. Inequality A.3 is equivalent to:

$$\sum_{i \in C_a(r) \cup C_d(r)} t_i(u_i(b_1) - u_i(b_2)) > 0.$$

Note that the set of indices $C_a(r) \cup C_d(r)$ is the set of all indices i for which $r \in R_i$. Thus, the above is equivalent to

$$\sum_{i : r \in R_i} t_i(u_i(b_1) - u_i(b_2)) > 0$$

and we can split the summation so we have

$$\sum_{i : r \in R_i} t_i u_i(b_1) > \sum_{i : r \in R_i} t_i u_i(b_2).$$

The above is the same as inequality 9.5, which is equivalent to inequality 9.4, which in turn is equivalent to inequality 9.3, and since we define an generalized additive utility function u as:

$$u(\vec{a}) = \sum_{i}^{Q} t_i u_i(\vec{a})$$

we can substitute this definition of $u(\vec{a})$ for the summations in inequality 9.3 to obtain:

$$u(b_1) > u(b_2)$$

whenever $(b_1, b_2) \models r$, which is the definition of consistency of u with r. □

Theorem 9.8 (Simple Scaling Parameters) *Given a set of feature sets $C = \{C_1, C_2, ..., C_Q\}$ and a set of feature vector statements M in $\mathcal{L}(A)$, with R_i the set of relevant statements for C_i, and each u_i is 1-consistent with each $r \in R_i$, then*

$$u(\vec{a}) = \sum_{i=1}^{Q} u_i(\vec{a}) \tag{A.4}$$

is an ordinal utility function consistent with M.

Proof. Fix a statement $r \in M$. Without loss of generality, assume all subutility functions are positively ordered. If each u_i is 1-consistent with R_i, then if $r \in R_i$, for all $(b_1, b_2) \models r$ we have $u_i(b_1) - u_i(b_2) > 0$. So we have

$$\sum_{i : r \in R_i} (u_i(b_1) - u_i(b_2)) > 0$$

and since for $r \notin R_i$ $u_i(b_1) = u_i(b_2)$ for all $(b_1, b_2) \models r$, we have:

$$\sum_{i=1}^{Q}(u_i(b_1) - u_i(b_2)) > 0$$

or

$$\sum_{i=1}^{Q} u_i(b_1) - \sum_{i=1}^{Q} u_i(b_2) > 0$$

Substituting $u(\vec{a})$ for the definition in equation A.4 we have

$$u(b_1) - u(b_2) > 0.$$

Since we chose r arbitrarily, this holds for any $r \in M$. Thus, u as defined in equation A.4 is a utility function consistent with M. □

Theorem 9.9 (Correctness of SAT-Formulation) *Given a set of attribute sets C of A, if $P(M, C)$ has no solution then the preferences M are not consistent with a utility function u of the form $u(\vec{a}) = \sum_i t_i u_i(\vec{a})$.*

Proof. If there is no solution to $P(M, C)$, it means that in all possible assignments of truth values to variables, some clause is not satisfied. By definition of the propositional variables in $P(M, C)$, the truth assignments correspond directly to statement-subutility function 1-consistency. The clauses in $P(M, C)$ are of two forms: statement-set clauses (definition 9.1), and conflict clauses (definition 9.2). By assumption, there is no solution to $P(M, C)$. Thus, in each possible assignment of truth values to the variables in $P(M, C)$, either there exists some statement-set clause that is not satisfied, or there exists some conflict clause that is not satisfied. In terms of the interpretation given to variables z_{ij}, if a statement-set clause is not satisfied then there is some statement r_j that is not 1-consistent with any subutility function u_i for each C_i. In the other case, if a conflict clause is not satisfied, then there exists some cycle Y_{ik} where each $r_j \in Y_{ik}$ is included in the set $\overline{R}_i$ (by definition of $\overline{R}_i$, equation 9.11); and thus $\overline{R}_i$ is not conflict-free.

Thus we must treat two cases. Suppose the first case holds: in any assignment of truth values to variables z_{ij} in $P(M, C)$, there is some statement r_j that is not 1-consistent with any subutility function u_i. Thus, by Corollary 9.3.2, a utility function $u(\vec{a}) = \sum_i t_i u_i(\vec{a})$ is not consistent with r_j. By definition of consistency, u is not consistent with M. This holds for any generalized additive utility function u based on the utility-dependence partition C.

The second case is that in any assignment of truth values to the variables z_{ij}, there is some conflict clause $C_{\overline{R}_i} = \vee_{j:r_j \in Y_{ik}} \neg z_{ij}$ where all z_{ij} are true. We show that this case is equivalent to the first case. By definition of $P(M, C)$, for all $r_j \in C_{\overline{R}_i}$, there is a statement-set clause $C_j = \vee_{i \in X_j} z_{ij}$. If there were some $z_{xj} \in C_j$ for $x \neq i$, where $z_{xj} = true$, we could construct a solution to $P(M, C)$ by making z_{ij} *false* and leaving z_{xj} *true*. But by assumption, there is no solution to $P(M, C)$, so we must conclude that for all $z_{xj} \in C_j$ with $x \neq i$, $z_{xj} = false$. Therefore, we make z_{ij} *false*, and then for all $z_{xj} \in C_j$, $z_{xj} = \mathit{false}$, so we now have an assignment of truth values of the first case.

This shows that if there is no solution to the SAT problem, $P(M, C)$, then there is no generalized additive utility function, consistent with the stated preferences M and the set of feature sets C. □

Theorem 9.10 (Set Merging) *If M is an inconsistent preference set over features C_k, then there does not exist any C_i, C_j with $C_i \cup C_j = C_k$ such that M restricted to C_i is consistent and M restricted to C_j is consistent.*

Proof. A set of preference M over C_k is inconsistent exactly when the model graph, $G(C_k)$ has one or more cycles. Let $Y = \{r_1, r_2, ..., r_k\}$ be such a cycle. Since Y is a cycle, there exists models $b_1, b_2, ..., b_k$ such that

$$\begin{array}{l} (b_1, b_2) \models r_1 \\ (b_2, b_3) \models r_2 \\ \vdots \\ (b_{k-1}, b_k) \models r_{k-1} \\ (b_k, b_1) \models r_k. \end{array}$$

Let $C \subseteq C_k$ be any set of features. Let C' be C if some $r \in Y$ has non-empty $(r \upharpoonright C)$, otherwise let $C' = C_k \backslash C$. In either case, some $r \in Y$ has non-empty $(r \upharpoonright C')$. By definition of model restriction and statement restriction, we have:

$$\begin{array}{l} ((b_1 \upharpoonright C'), (b_2 \upharpoonright C')) \models (r_1 \upharpoonright C') \\ ((b_2 \upharpoonright C'), (b_3 \upharpoonright C')) \models (r_2 \upharpoonright C') \\ \vdots \\ ((b_{k-1} \upharpoonright C'), (m_k \upharpoonright C')) \models (r_{k-1} \upharpoonright C') \\ ((b_k \upharpoonright C'), (b_1 \upharpoonright C')) \models (r_k \upharpoonright C'). \end{array}$$

Thus we have exhibited a cycle on C' caused by the preferences M. Let $C_i = C'$, $C_j = C_k \backslash C'$. Then we have shown, for any C_i, C_j such that $C_i \cup C_j = C_k$, that there is a conflict in C_i. □

Lemma 9.3.4 (Sufficiency of $I(M, C, \overline{R})$) *Given $I(M, C, \overline{R})$ and a solution to it providing an assignment of values to $t_1, ..., t_Q$, for all r in some $(R_i \backslash \overline{R_i})$ and for all basket pairs $(b_1, b_2) \models r$,*

$$\sum_{i \in \{C_a(r) \cup C_d(r)\}} t_i(u_i(b_1) - u_i(b_2)) > 0.$$

Proof. Let x_i be the maximum value of $u_i(b_1) - u_i(b_2)$ and y_i be the minimum value of $u_i(b_1) - u_i(b_2)$, then the following inequalities are members of $I(M, C, \overline{R})$:

$$\begin{array}{l} x_i t_i + \sum_{j=1, j \neq i}^{Q} t_j x_j > 0 \\ x_i t_i + \sum_{j=1, j \neq i}^{Q} t_j y_j > 0 \\ y_i t_i + \sum_{j=1, j \neq i}^{Q} t_j x_j > 0 \\ y_i t_i + \sum_{j=1, j \neq i}^{Q} t_j y_j > 0. \end{array}$$

Since we have a solution to $I(M, C, \overline{R})$, we have an assignment of values to $t_1, ..., t_Q$ such that the above hold. Clearly the following hold for any $(u_i(b_1) - u_i(b_2))$ with $x_i \geq (u_i(b_1) - u_i(b_2)) \geq y_i$:

$$(u_i(b_1) - u_i(b_2))t_i + \sum_{j=1, j\neq i}^{Q} t_j x_j > 0$$
$$(u_i(b_1) - u_i(b_2))t_i + \sum_{j=1, j\neq i}^{Q} t_j y_j > 0.$$

Since this holds for any choice of $i : 1 \geq i \geq Q$, this proves the lemma. $\square$

Theorem 9.11 (Qualitative Preference Inequalities) *If the system of linear inequalities, $I(M, C, \overline{R})$, has a solution, this solution corresponds to a utility function u consistent with M.*

Proof. Each subutility function, u_i, $1 \leq i \leq Q$, is fixed. By Lemma 9.3.4, for a given statement r in some $(R_i \backslash \overline{R}_i)$, we are assured that if $(b_1, b_2) \models r$ then $\sum_i t_i u_i(b_1) > \sum_i t_i u_i(b_2)$, for all $i \in \{C_a(r) \cup C_d(r)\}$. Since $C_a(r)$ and $C_d(r)$ are disjoint, we can rearrange the summations and obtain:

$$\sum_{i \in C_a(r)} t_i(u_i(b_1) - u_i(b_2)) > \sum_{j \in C_d(r)} t_j(u_j(b_2) - u_j(b_1)).$$

Then by Theorem 9.7, u is consistent with r.

If r is not in any $(R_i \backslash \overline{R}_i)$, then for all i such that $r \in R_i$, u_i is 1-consistent with r. For j such that $r \notin R_j$, $(b_1, b_2) \models r$ implies that $(b_1 \upharpoonright S_j) = (b_2 \upharpoonright S_j)$, so all u_j are 1-consistent with r. Since all subutility functions are 1-consistent with r, we have:

$$\sum_{i}^{Q} t_i(u_i(b_1) - u_i(b_2)) > 0$$

and u is consistent with r. Thus, u is consistent with all r in some $(R_i \backslash \overline{R}_i)$, and all r not in some $(R_i \backslash \overline{R}_i)$, so u is consistent with all $r \in M$. This proves the lemma. $\square$

Bibliography

[AGLM04] Silvia Angilella, Salvatore Greco, Fabio Lamantia, and Benedetto Matarazzo. Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research*, 158:734–744, 2004.

[BB91] Jane Beattie and Jonathan Baron. Investigating the effect of stimulus range on attribute weight. *Journal of Experimental Psychology: Human Perception and Performance*, 17(2):571–585, 1991.

[BBB01] Craig Boutilier, Fahiem Bacchus, and Ronen L. Brafman. Ucp-networks: A directed graphical representation of conditional utilities. In *Proceedings of Seventeenth Conference on Uncertainty in Artificial Intelligence*, Seattle, 2001. To Appear.

[BBGP97] Craig Boutilier, Ronen Brafman, Christopher Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In Jon Doyle and Richmond H. Thomason, editors, *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 19–28, Menlo Park, California, 1997. AAAI.

[BBHP99] Craig Boutilier, Ronen I. Brafman, Holger H. Hoos, and David Poole. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of Uncertainty in Artificial Intelligence 1999 (UAI-99)*, 1999.

[BD02] Ronen I. Brafman and Carmel Domshlak. Introducing variable importance tradeoffs into cp-nets. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*. Morgan Kaufmann, August 2002.

[BDK04] Ronen I. Brafman, Carmel Domshlak, and Tanya Kogan. Compact value-function representations for qualitative preferences. In *Proceedings of Uncertainty in Artifical Intelligence (UAI'04)*, pages 51–59, Arlington, Virginia, 2004. AUAI Press.

[BG95] Fahiem Bacchus and Adam Grove. Graphical models for preference and utility. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 3–19. Morgan Kaufmann, 1995.

[BG96] Fahiem Bacchus and Adam Grove. Utility independence in a qualitative decision theory. In *Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning*, pages 542–552. Morgan Kaufmann, 1996.

[BLKL01] M. Bichler, J. Lee, C. H. Kim, and H. S. Lee. Design and implementation of an intelligent decision analysis system for e-sourcing. In *Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI'2001), Las Vegas, Nevada*, pages 84–89, 2001.

[Bou94a] Craig Boutilier. Toward a logic for qualitative decision theory. In Jon Doyle, Erik Sandewall, and Piero Torasso, editors, *KR94*, San Francisco, may 1994. Morgan Kaufmann.

[Bou94b] Craig Boutilier. Toward a logic for qualitative decision theory. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-94)*, pages 76–86, Bonn, 1994.

[Bre04] Gerhard Brewka. Answer sets and qualitative decision making. *Synthese, special issue on Non-Monotonic and Uncertain Reasoning in the Focus of Paradigms of Cognition*, 2004. in press.

[BZ04] Marko Bohanec and Blaž Zupan. A function-decomposition method for development of hierarchical multi-attribute decision models. *Decision Support Systems*, 36:215–233, 2004.

[Chv83] Vašek Chvátal. *Linear Programming.* W.H. Freeman and Company, New York, 1983.

[CKO01] U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *18th International Conference on Machine Learning (ICML '01)*, pages 35–42, 2001.

[DFP97] Didier Dubois, Helene Fargier, and Henri Prade. Decision-making under ordinal preferences and comparative uncertainty. In *Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 157–164, 1997.

[DJ05] C. Domshlak and T. Joachims. Unstructuring user preferences: Efficient non-parametric utility revelation. In *Proceedings of the Twenty-first Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 169. AUAI press, 2005.

[DLBPS98] D. Dubois, D. Le Berre, H. Prade, and R. Sabbadin. Logical representation and computation of optimal decisions in a qualitative setting. In *Proceedings of AAAI-98*, pages 588–593, Menlo Park, California, 1998. AAAI Press.

[DSW91] Jon Doyle, Yoav Shoham, and Michael P. Wellman. A logic of relative desire (preliminary report). In Zbigniew Ras, editor, *Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems*, Lecture Notes in Computer Science, Berlin, 1991. Springer-Verlag.

[DT99] Jon Doyle and Richmond H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68, Summer 1999.

[EB94] W. Edwards and F. Hutton Barron. Smarts and smarter: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Making*, 60:306–325, 1994.

[Edw77] W. Edwards. How to use multiattribute utility analysis for social decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:326–340, 1977.

[Fis64] Peter C. Fishburn. *Decision and Value Theory.* John Wiley & Sons, New York, 1964.

[Fis95] Gregory W. Fischer. Range sensitivity of attribute weights in multiattribute value models. *Organizational Behavior and Human Decision Processes*, 62(3):252–266, June 1995.

[Gra96] Michel Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89:445–456, 1996.

[Han89] Sven Ove Hansson. A new semantical approach to the logic of preference. *Erkenntnis*, 31:1–42, 1989.

[Han96] Sven Ove Hansson. What is ceteris paribus preference? *Journal of Philosophical Logic*, 25:307–332, 1996.

[HQ80] James M. Henderson and Richard E. Quandt. *Microeconomic Theory: A Mathematical Approach.* McGraw-Hill, New York, third edition, 1980.

[JLS82] E. Jacquet-Lagreze and J. Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the uta method. *European Journal of Operational Research*, 10:151–164, 1982.

[Kee92] Ralph L. Keeney. *Value-focused thinking: a path to creative decision making.* Harvard University Press, Cambridge, Massachusetts, 1992.

[KR76] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Wiley and Sons, New York, 1976.

[LS99] Piero La Mura and Yoav Shoham. Expected utility networks. In *Proc. of 15th conference on Uncertainty in Artificial Intelligence*, pages 366–373, 1999.

[LvdTW02] J. Lang, L. van der Torre, and E. Weydert. Utilitarian desires. *Autonomous Agents and Multi-Agent Systems*, 5:329–363, 2002.

[MD02] Michael McGeachie and Jon Doyle. Efficient utility functions for ceteris paribus preferences. In *AAAI Eighteenth National Conference on Artificial Intelligence*, Edmonton, Alberta, August 2002.

[MD04] Michael McGeachie and Jon Doyle. Utility functions for ceteris paribus preferences. *Computational Intelligence: Special Issue on Preferences*, 2004. Submitted 2003.

[MHS05] J. Mustajoki, R. P. Hämäläinen, and A. Salo. Decision support by interval smart/swing-incorporating imprescision in the smart and swing methods. *Decision Sciences*, 36(2):317–339, 2005.

[MPY92] Herbert Moskowitz, Paul V. Preckel, and Aynang Yang. Multiple-criteria robust interactive decision analysis (mcrid) for optimizing public policies. *European Journal of Operational Research*, 56:219–236, 1992.

[P9̈9] Mari Pöyhönen. *On attribute weighting in value trees.* PhD thesis, Helsinki University of Technology, April 1999.

[PH01] M. Poyhonen and R.P. Hamalainen. On the convergence of multiattribute weighting methods. *European Journal of Operational Research*, 129(3):569–585, March 2001.

[Rai68] Howard Raiffa. *Decision Analysis: Introductory lectures on choices under uncertainty.* Addison-Wesley, Reading, Massachusetts, 1968.

[Rib96] Rita A. Ribeiro. Fuzzy multiple attribute decision making: A review and new preference elicitation techniques. *Fuzzy Sets and Systems*, 78:155–181, 1996.

[Saa80] Thomas L. Saaty. *The analytic hierarchy process.* McGraw Hill, New York, 1980.

[Sav54] Leonard J. Savage. *The Foundations of Statistics.* John Wiley & Sons, New York, 1954.

[SH92] Ahti A. Salo and Raimo P. Hamalainen. Preference assessment by imprecise ratio statements. *Operations Research*, 40(6):1053–1061, 1992.

[SIT04] Kazutomi Sugihara, Hiroaki Ishii, and Hideo Tanaka. Interval priorities in ahp by interval regression analysis. *European Journal of Operational Research*, 158:745–754, 2004.

[SLM92] Bart Selman, Hector J. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In Paul Rosenbloom and Peter

Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California, 1992. AAAI Press.

[SW84] Andrew P. Sage and Chelsea C. White. Ariadne: A knowledge-based interactive system for planning and decision support. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(1):35–47, 1984.

[Tha99] Richard H. Thaler. Mental accounting matters. *Journal of Behavioral Decision Making*, 12:183–206, 1999.

[TK86] Amos Tversky and Daniel Kahneman. Rational choice and the framing of decisions. *Journal of Business*, 59:251–278, 1986.

[TP94] Sek-Wah Tan and Judea Pearl. Qualitative decision theory. In *AAAI94*, Menlo Park, CA, July 1994. AAAI Press.

[vdTW98] L. van der Torre and E. Weydert. Goals, desires, utilities and preferences. In *Proceedings of the ECAI'98 Workshop on Decision Theory meets Artificial Intelligence*, 1998.

[vdTW01] L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory. *Apllied Intelligence*, 14:285–301, 2001.

[vNM44] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[vWE86] Detlof von Winterfeldt and Ward Edwards. *Decision Analysis and Behavioral Research.* Cambridge University Press, Cambridge, England, 1986.

[WD91] Michael Wellman and Jon Doyle. Preferential semantics for goals. In Thomas Dean and Kathleen McKeown, editors, *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 698–703, Menlo Park, California, 1991. AAAI Press.

[Web85] M. Weber. A method of multiattribute decision making with incomplete information. *Management Science*, 31(11):1365–1371, 1985.

[Wel90] Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

[Wil04] Nic Wilson. Extending cp-nets with stronger conditional preference statements. In *Proceedings of Nineteenth National Conference on AI (AAAI'04)*, pages 735–741, 2004.

[ZBDB99] Blaž Zupan, Marko Bohanec, Janez Demšar, and Ivan Bratko. Learning by discovering concept hierarchies. *Artificial Intelligence*, 109:211–242, 1999.