T 1673

MIT/LCS/TR-340

REASONING ABOUT PREFERENCE MODELS

Michael Paul Wellman

Reasoning About Preference Models

by

Michael Paul Wellman

May 1985

This research was supported (in part) by the National Institutes of Health Grant No. NIH-5-R24-RR01320-02.

Reasoning About Preference Models

by
Michael Paul Wellman

This report is a modified version of a thesis submitted to the Department of Electrical Engineering and Computer Science on May 21, 1985 in partial fulfillment of the requirements for the Degree of Master of Science

Abstract

Programs that make decisions need mechanisms for representing and reasoning about the desirability of the possible consequences of their choices. This work is an exploration of *preference models* based on utility theory. The framework presented is distinguished by a *qualitative* view of preferences and a *knowledge-based* approach to the application of utility theory. The design for a comprehensive preference modeler is implemented in part by the *U*tility *R*easoning *P*ackage (URP), a collection of facilities for constructing and analyzing preference models.

Qualitative mathematical reasoning techniques are employed to develop *partial* specifications of single-attribute utility functions from qualitative preference assertions. Functions are described in terms of gross behaviors, symbolic forms, and parametric constraints. Appropriate dominance-testing algorithms are chosen from a knowledge base of stochastic dominance routines based on qualitative properties of the utility function. URP constructs multiattribute utility functions from a set of independence conditions by applying proof rules from a knowledge base containing the important decomposition theorems from the literature. Proof rules describe the logical relations among independence conditions and functional forms. Hierarchical decompositions are structured automatically.

Flexible model construction provides the potential for interpreting preference choices under a procedure that does not depend on the underlying utility model. Model-independent interpretation enables assessment under a wide range of descriptive theories of preference choice. Domain-specific preference knowledge is incorporated in URP by tying domain concepts to the modeler's *technical vocabulary.* Development of a health preference knowledge base illustrates how preference modeling could be included in a knowledge-based system for a particular application.

Thesis Advisor: Peter Szolovits
Title: Associate Professor of Computer Science

Keywords: utility theory, decision making, mathematical modeling, preferences, qualitative reasoning, knowledge-based systems, multiattribute utility, stochastic dominance, utility assessment, medical decision making

Acknowledgements

Alvin Drake and Ralph Keeney for some early advice and encouragement.

David Bell for much of the same and for looking over part of the knowledge base.

Members, past and present, of the Division of Clinical Decision Making at Tufts-New England Medical Center for two years of lively case discussions and for putting up with my medical ignorance. Special thanks to Van Dunn for the medical example, and to Jim Hollenberg for helping me select and set up the case as well as for many fruitful discussions about AI and Decision Analysis. And to Stephen Pauker for general tutelage, and for keeping me somewhat closer to the real world.

Kathe Unrath for eliminating some of the stylistic monstrosities from this document.

The members of the Clinical Decision Making Group at MIT for lots of good discussions, stimulating ideas, and in general for providing a nice place to work. Particular thanks to Zak Kohane for many heated discussions about this project and for helpful suggestions on several drafts, to Elisha Sacks for draft-reading and for helping me to get the QM package to do what I needed, and to Robert Kunstaetter for tolerating my naive medical questions and some of my more outrageous arguments.

Peter Szolovits, my advisor, for continued support and guidance.

Table of Contents

List of Figures

List of Tables

1. Introduction

The work described in this thesis is an investigation of an approach toward representing and reasoning about preferences in a machine. These preferences "belong" to some agent (person, institution, society) called the *decision maker*. We wish to encode them in a computer program so that it may be more effective in helping the agent make decisions, or even in making decisions for the agent by proxy.

Decision scientists have long been concerned with formal models of preference, and have assembled over the years an impressive body of theory which bears directly on the enterprise of representing preferences in computers. Attempts to employ these models in decision-making computer programs, however, have met with only limited success, mainly because the implementations have incorporated a highly restricted view of the formal preference theories. Taking such a view is difficult to avoid using conventional programming strategies.

This project arose out of the observation that utility theory (like many mathematical modeling disciplines) is tremendously richer and remarkably more flexible than reflected in any program performing utility analysis. This observation, together with advances in knowledge-based programming technology and qualitative reasoning methods from artificial intelligence, suggests the approach related in this thesis.

1.1 Preferences

People make decisions to influence the course of events in ways that they deem desirable. Different actions make the various possible outcomes more or less likely to obtain; choosing among available acts is an exercise in selecting probable futures. It seems important, then, that computer programs that make decisions or help humans to do so should have some means to reason about the desirability of the possible consequences of these decisions.

Mainstream AI research—while enormously concerned with representing beliefs and states of knowledge—has for all practical purposes ignored the issue of preferences. Note that goals as employed by AI planning programs do not function as expressions of preference; at the very best they can be said to name the most desired outcome. More often, goals are used to describe intentions or desired reasoning states, which are nothing at all like desired states of the world.[1] Furthermore, in most real decision-making or planning situations it is necessary to consider possible outcomes other than the one most preferred. The program must be able to compare the desirability of various partially satisfied or uncertain goals, which may not be related to the absolute goal by any identifiable distance metric.

1.2 Preference Models

This work is an exploration of the use of *preference models* as a basis for reasoning about desires in decision making. Let us define a *prospect* to be a probability distribution over consequences, or *outcomes*. The underlying assumption of preference modeling is that there exists a *preference relation*, denoted by $\succ$, which orders the set of prospects. A preference model is a structure used to represent this order relation.

Preference models can be used to answer questions of the form "Is x preferred to y?" where x and y are prospects relevant to the decision being considered. The decidability of such questions depends on the degree to which the preference model is specified. In general, a preference model will characterize only a subset of the true preference relation it represents. Thus, any partial description of a preference order may be termed a preference model.

The ultimate goal of this effort is to develop a representation for preference models that will accommodate arbitrary amounts and types of information about the preference relation, and a reasoning capacity that will enable us to reconstruct as much as possible of the relation from whatever information is available. A formal characterization of the sorts of information that may be incorporated in preference models and the language used to describe it may be called a *preference calculus*. While I cannot claim to provide a sufficiently formal or principled development, this work may be viewed as a step toward the definition of a useful preference calculus.

The next major step (which requires a leap of faith, since I do not intend to provide detailed justification here) is that the foundations for this preference calculus are to be found in utility theory. A fundamental result of this theory is that, given a few appealing axioms,[2] there exists a *utility function* which maps prospects to real numbers (called *utilities*) in such a way that the utility of any prospect is equal to

[1] Doyle [23] discusses the importance of distinguishing between desires and intentions in a decision making program.

[2] By appealing, I mean that they generally express properties that one would like to conform to in

the sum of the utility of each certain outcome weighted by its probability in that prospect. Utilities rank the prospects by preference; $u(p_1) > u(p_2)$ if and only if $p_1 \succ p_2$. If we base our preference model on the notion of a utility function, we can exploit a large body of results from utility theory, and more general analytical techniques from the mathematics of functions. This thesis is largely an account of some of these possible exploitations.

Formally, a utility function u maps states of the world (outcomes) x in the *outcome space* $X = X_1 \times \cdots \times X_n$ to real numbers in such a way that the optimal decision is the one that maximizes the expected value of this function. Often x is expressed as a vector of attributes, $(x_1, x_2, \ldots, x_n), x_i \in X_i$, where the values of the x_is totally describe the aspects of the outcome pertinent to the decision.

While a major preference modeling effort is often undertaken for the sake of the *insight* it brings to decision making, I will be primarily concerned with the use of preference models for the analysis of a particular decision.[3] For our purposes, a decision involves a choice among a set of strategies, each of which is associated with some resulting outcome $x \in X$ (or more generally, a prospect $p \in P$, where P, the *decision space* is the set of simple probability distributions over X). To make the decision, we must figure out which strategy has the greatest expected utility.

1.3 The Basic Approach

The main features that distinguish the attitude toward utility modeling developed here from traditional computerized utility modeling are its qualitative view of preference structures and its knowledge-based approach to the encoding of utility-theoretic concepts. Taken together, these features result in a system which emphasizes model structuring over model parameterization. This attitude has been embodied in a collection of programs and overall framework for preference model-

decision making. This is not to say that these axioms are totally accepted. However, it is not my intention to enter the debate at this point. A particularly lucid axiomatic development of expected utility can be found in Savage's important work [92]. Alternative axiomatizations are provided by Fishburn [35]. The original existence proof is due to von Neumann and Morganstern [107].

[3] Although, as will be explained later, I believe that models created using this methodology offer the potential for greater insight due to their deep, explicit qualitative structure. And there is no reason to believe that any of the insight-yielding characteristics that are inherent in traditional utility analysis efforts will not carry over to this methodology. Unfortunately, I do not know how to prove an insight theorem.

ing called URP (for *U*tility *R*easoning *P*ackage). The sections immediately below describe features of the general attitude; descriptions of the actual URP framework and programs make up the remainder of this thesis.

1.3.1 Qualitative View of Preferences

An important novelty of the approach described here is in the qualitative mechanisms used to analyze preference structures. Utility analysis is usually considered a primarily quantitative[4] task, and previous utility modeling aids (basically assessment tools) have reflected that bias. I believe that the emphasis on numerical computation has tended to obscure the more important modeling and analysis issues, just as implementations of mathematical models often hide the central qualitative structure of the problems they are supposed to represent. Recent work in artificial intelligence (that of Kuipers [67], Forbus [39], and Sacks [90], for example) has shown that it is often possible to reason effectively about the qualitative behaviors of mathematical structures without resorting to overly precise descriptions. Such a capability would be enormously useful for utility analysis, because (as with any kind of psychological measurement) virtually any numeric data is uncomfortably precise. A more complete account of the potential advantages is given in section 9.1.2.

This qualitative view meshes nicely with the notion of a preference model as a *partial* description of a preference order. The view is supported by URP in its tendency to focus on propositions about the utility structure rather than on direct specification of components making up the structure. Specific methods contributing to this focus are described below.

1.3.2 Knowledge-Based Approach to Utility Theory

The second major distinction in how URP models preferences is that it relies on utility theory knowledge expressed in a (relatively) declarative format, rather than compiled into analysis procedures. Encoding utility-theoretic concepts in explicit knowledge structures leads to advantages in flexibility, extensibility, and explain-

[4]It is difficult to avoid the confusion inherent in words like *qualitative* and *quantitative*. Naturally, all mathematical models are composed of relationships among quantities, and in that sense they are quantitative. But it is important to distinguish qualitative *descriptions* of quantities from numeric descriptions. A symbolic expression is just one form of qualitative description. The term *qualitative* as used here is almost always meant in contrast to *numeric*, or more generally in opposition to excess precision of any sort.

ability, which will be noted as we get into specifics. For now I will just mention that the process of building URP has resulted in the collection of a large corpus of utility-theoretic knowledge, encoded in a uniform representation, *executable* by URP's interpreter. The creation of this knowledge base (especially that pertaining to multiattribute decomposition) may be a worthwhile exercise in its own right.

1.4 Model Construction and Analysis

A high-level view of the URP preference modeling framework is depicted in figure 1.1. Given a probabilistic model of the alternative strategies, the program will produce decisions based on a specification of preferences. The probabilistic model is given as $(p_1, \ldots, p_m)$, where p_i, the result of the ith strategy, is a prospect in P (that is, a simple probability distribution over the outcome space, X).

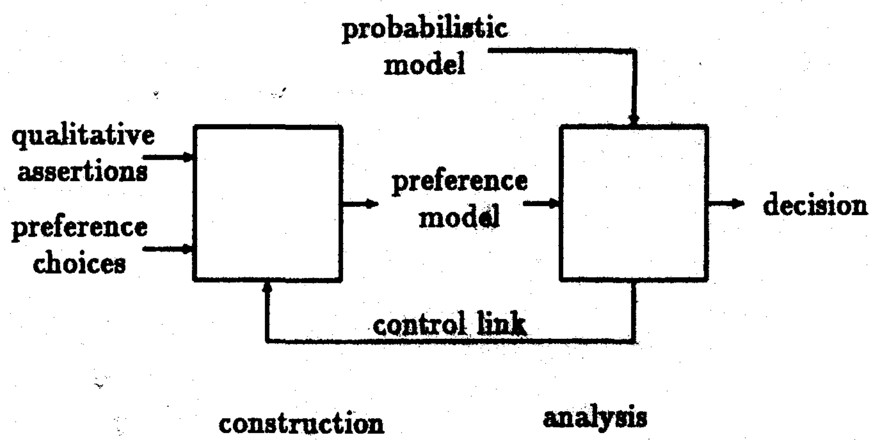


Figure 1.1: A high-level view of the program's behavior

Preference specifications are in terms of the same basic variables as the probabilistic model: the attributes $X_1, \ldots, X_n$. The two main categories of specifications are qualitative utility assertions and preference choices. Qualitative utility assertions are propositions that refer to preference conditions for attributes or sets of attributes. Some illustrative examples, stated in English (rather than URPese[5]), with the js and Ys standing for attributes and attribute sets, respectively:

[5]It will soon be demonstrated that URPese is closely related to the technical language of utility theory.

- Preference for prospects over attributes in Y_1 is independent of the fixed value of the rest of the attributes
- All other things equal, higher levels of (numeric) attribute j_1 are preferred
- The attributes in Y_2 are complementary
- The marginal value of equal increments of attribute j_2 is decreasing with j_2
- An additive form is valid for the utility function over Y_3

The implications of these assertions on the preference model will be made clearer below.

Preference choices are assertions similar to those generated by the standard lottery technique of utility assessment, where the assessor is required to report preferences in hypothetical choices among simplified prospects. The form and use of preference choices is the subject of chapter 7.

The output of the first black box of figure 1.1 is a data structure representing the current preference model. The model is a collection of assertions about the utility structure, containing some combination of functional constraints, algebraic forms, parametric constraints, and other qualitative properties. Because the model is the central object of reasoning activity, it is useful to separate the process into two components: *model construction* and *model analysis*. This is a natural distinction, because the two activities use very different types of knowledge and inference mechanisms. The processes are not totally independent, of course, since we construct models in the first place for purposes of analysis. Therefore, there must be a control link tying the results of the analysis to the goals of the construction.

1.4.1 Model Construction

Several mechanisms are employed to construct preference models from the qualitative properties and preference choices input. Of course, by my definition these raw input assertions alone can constitute a preference model, but we are typically more interested in higher-level inferences we can make from these premises.

URP's most completely developed model construction mechanisms deal with assertions about *independence axioms* and *qualitative behaviors* of single-attribute utility functions. Examples of URP's reasoning in these areas are given in the next chapter; deeper analyses of these mechanisms appear in chapters 3 and 5.

1.4.2 Model Analysis

The goal of the model analysis module is to determine which of the available strategies yields the highest expected utility given a probabilistic model and a preference model. This will not always be possible, either because the expressions for expected utility are underconstrained, or because the analyzer is not powerful enough to recognize the dominance of an expression.[6] If the procedure succeeds in proving that one alternative is optimal, the process is finished. Otherwise, it is necessary to refine the preference model. Examples of refinement goals are to constrain a certain parameter or set of parameters, to determine the applicability of a functional constraint such as monotonicity or concavity, or to find a simpler utility function that has fewer parameters.

1.5 Status of the Implementation

This thesis report describes both a design for a comprehensive preference modeler and an implementation that realizes part of that design. I use "URP" to refer to both of these at times; where it may be unclear I explicitly distinguish the *implemented* URP from the URP *framework*. I have taken the implementation to a point that demonstrates the feasibility and potential of this approach, and at the same time provides some interesting and useful capabilities. Throughout this thesis I point out parts of the implemented URP that need to be extended, and discuss ways in which the program could be improved.

Development of the complete preference modeling system as envisioned here could potentially require an effort similar in size to the MACSYMA project. Because the system revolves around a task rather than a particular method, a comprehensive program would be composed of a multitude of separate mechanisms and representations. Indeed, even the implemented portion of URP includes several distinct components. Coordinating the use of these separate programs for a central task is a considerable engineering problem in itself.

The usefulness of the implemented part is illustrated by the example of chapter 2. There it is shown that the program is able to reduce the set of admissible options from those considered in an actual medical decision based on weak, qualitative assumptions about the patient's preference structure. Various kinds of reasoning

[6] One expression *dominates* another if its value is greater for any allowable assignment of the parameters.

about the qualitative properties themselves and how they relate to the utility function are also demonstrated. Though the program cannot always take advantage of the inferences it makes about the function for decision making, a decision analyst would find these facilities useful for developing utility models that could then be used in conventional decision analysis tools (along the lines of the program described by Pauker and Kassirer [82], in routine use at Tufts-New England Medical Center).

The facilities implemented for multiattribute decomposition are limited to model-structuring. To use a multiattribute model generated by URP for an actual analysis, one would have to transfer the model to a package that performs multiattribute assessment and evaluation.[7] Utility theorists may also find the program useful for exploring novel decompositions based on different combinations of axioms. One can test the implications of new results in multiattribute utility by encoding them as URP independence concepts and theorems, adding them to the existing knowledge base.

The main objective of this work has been to demonstrate that the overall approach to preference modeling is feasible and worthwhile. An assessment of how this project has satisfied that goal is deferred to the concluding section.

1.6 Guide to the Thesis

This introductory chapter has served as an abstract description of the preference modeling task, and as an overview of the approach to preference modeling embodied by this project. The rest of this thesis describes the URP preference modeling framework in greater detail: including working representations and mechanisms as well as unimplemented components that have been worked out to some degree.

Chapter 2 provides the setting for the rest of the thesis by describing a medical decision example and illustrating some preference modeling capabilities of URP. The case exemplifies the kind of decision problem of interest: a situation where there are several feasible strategies, none of which is obviously the best course of action. The example also serves to provide a context for the more detailed descriptions of URP's mechanisms that follow.

[7]For example, Keeney and Sicherman's MUFCAP [62] can accept as input many of the models that might be constructed by URP. Though using a conventional utility analysis package sacrifices some of the advantages of the URP framework (the ability to reason about partially specified functions, for example), there is still substantial benefit to structuring the model using URP.

The next chapter describes the first of two major preference modeling mechanisms implemented in URP: reasoning about the qualitative behavior of single-attribute utility functions. URP's mechanisms for relating these behaviors to the mathematical structure of the utility function is discussed, with emphasis on qualitative mathematical reasoning methods employed. The utility-theoretic content of URP's single-attribute knowledge is delimited. Deriving decisions based on URP's representation of single-attribute functions is the subject of chapter 4, which also includes a discussion of dominance-testing procedures incorporated into the URP framework.

The following two chapters deal with URP's facilities for multiattribute decomposition. Chapter 5 describes the program's use of independence conditions to derive multiattribute functional forms, based on proofs of their validity generated from a knowledge base of decomposition theorems. This chapter describes the theorem language as well as the interpreter that uses the proof rules to derive the functional forms. The knowledge base itself is described in chapter 6, an in-depth discussion of the multiattribute utility theory known to URP. There I describe the concepts appearing in the knowledge base, and provide rationale for their inclusion. I also argue that the knowledge base covers a substantial portion of the important decomposition results developed by utility theorists.

In chapter 7 I discuss how assessment based on hypothetical preference choices—the mainstay of traditional utility analysis programs—may be incorporated in the URP framework. The major point of the chapter is how URP's flexible model structuring capabilities and capacity for reasoning about incompletely specified models leads to less restrictive assessment. Of particular interest are the ability to work with ad hoc constraints, and the intriguing notion of interpretation under descriptive theories of preference choice.

Representation of domain-specific preference knowledge is the subject of chapter 8. There I argue that concepts from a domain-independent modeling system must be encoded in a technical vocabulary, and therefore are not necessarily intuitive for modeling in particular application domains. This problem is remedied by the construction of a domain/modeling concept mapping, in the form of a knowledge base. The chapter includes a substantial development of a knowledge base of health preferences, encoded in URP's technical vocabulary.

The concluding chapter (9) contains the usual stuff of which conclusions are made: limitations, comparison with related work, future directions, and a summary of contributions. The contributions are partitioned into three areas, each speaking to a different (though possibly overlapping) potential audience for this work:

1. Preferences in decision making programs
2. Utility analysis
3. Expert systems for mathematical modeling

Skipping ahead to the concluding chapter (especially the perspectives and contributions sections) before reading much further in the thesis is recommended for those in need of further motivation for undertaking this approach to preference modeling.

Finally, the URP knowledge base is presented in the appendices. The knowledge base specification, in conjunction with informal procedural descriptions from the body of this document, should provide sufficient information to reproduce (and improve upon) the preference modeling capability described herein.

2. A Medical Decision Example

In this chapter I present the outline of a medical decision problem that typifies the kind of decision to which URP might be applied. After setting up the problem, I work through some exercises in preference modeling that could be performed in an application of URP to this problem. Remember that the purpose of this exercise is to demonstrate URP's capabilities; I am not making suggestions about the patient's actual preferences in this case. It is not at all important to understand the medical context of this decision to appreciate the preference modeling facilities. Therefore, my description of the case focuses on the structure of the problem at the expense of (clinically important) details of the medical situation.

2.1 The Decision Problem

The case is taken from a clinical decision consult performed by Dunn at the Division of Clinical Decision Making at Tufts-New England Medical Center [24]. It is a representative example of the many routine clinical decision analyses performed by members of that group [22] over the past eight years.

2.1.1 Case Description

The patient is a 72-year-old white male with a large abdominal aortic aneurysm (AAA) and a history of coronary artery disease (CAD) and cerebrovascular disease (CVD). There are potential treatments for each of these problems: aneurysm resection (removal) for the AAA, coronary artery bypass graft (CABG) surgery to fix the CAD, and carotid endarterectomy to relieve the CVD. His most pressing problem is the aneurysm; without surgery to repair the AAA his prognosis is poor. However, his underlying CAD and CVD place him at a greatly increased risk for death or disabling stroke during the AAA surgery. The question is whether to go ahead and operate to repair the aneurysm anyway, or to try and fix the patient's coronary artery and carotid problems first. Five primary options were considered in the analysis:

1. **no-repair**: Do nothing.

2. **aaa**: Perform surgery to repair the aneurysm only.

3. **cabg**: Perform cardiac catheterization to determine the extent and operability of CAD. Perform CABG, if indicated, to reduce AAA operative mortality. Then resect the aneurysm.

4. **endart**: Perform carotid arteriography to determine the severity of CVD. If severe and operable, perform carotid endarterectomy to reduce likelihood of disabling stroke during AAA surgery. Then resect the aneurysm.

5. **both**: Perform both tests and whichever procedures (CABG and/or endarterectomy) are indicated, followed by aneurysm resection.

In addition, two variations were identified: **cabg*** and **both***, which differ from their counterparts above in their criteria for performing CABG. In the variation strategies, bypass surgery is not performed for three-vessel CAD.

2.1.2 The Analysis

In the original analysis, the decision criterion was expected survival, with disabling stroke treated like immediate death. A large probability tree was constructed, resulting in a probability distribution over expected lifetimes (a prospect) for each strategy. The expectation for years of life for each prospect is given in table 2.1.

strategy	life expectancy (years)
no-repair	2.7
aaa	5.1
cabg	7.0
endart	5.2
both	7.4
cabg*	6.2
both*	6.5

Table 2.1: Baseline life expectancies for each strategy

Survival was computed using the DEALE model of life expectancy [2]. Each terminal node in the decision tree is associated with a DEALE parameter, which approximates the life expectancy for that outcome. For now and in the remainder of this chapter I will treat this parameter as if it represented the actual lifetime of the patient for the outcome in question; the implications of that treatment and a further discussion of the DEALE model is deferred to section 8.6.1.

The analysis also included a substantial amount of sensitivity analysis and discussion, concluding that **cabg** and **both** were the preferred strategies, with **both** offering a marginal advantage over **cabg**.

2.2 Qualitative Reasoning About Single-Attribute Utility

For the remainder of this chapter, I will take the baseline probabilistic model developed for this case as given, and use URP to explore a variety of preference modeling issues. The outcome space (X) is the set of values possible for *life-years*:[1] the positive real numbers, perhaps bounded by a practical maximum for this patient. For disabling stroke, x is zero regardless of lifetime.

The preference model used in the original analysis is simply maximization of life expectancy, $u(x) = x$. This function implies that all prospects offering the same expected value for life years are equally preferred. For example, one would be indifferent between living five years for certain and flipping a fair coin for ten years versus immediate death. We can apply this model using URP, but it would be much more interesting to explore the range of less-restrictive models that the program is able to generate and reason about. Though less-restrictive models will in general yield fewer conclusions, the conclusions that they do imply are based on weaker assumptions.

Construction of single-attribute utility functions often begins with the specification of qualitative behaviors of the function. In this case, an obvious qualitative property of preference for *life-years* is that "more is better." In other words, the utility function is increasing in *life-years*. This can be represented in URP via the assertion

(monotonic-increasing *life-years*)

This assertion, although based on trivial intuition, tells us quite a bit about the utility function. The outcomes (under certainty) are now totally ordered with respect to preference, and it is even possible to rank some nontrivial uncertain prospects

[1] Even though the name of the outcome attribute is "*life-years*," I will continue to use "x" in mathematical expressions for notational convenience.

(as we will see in the next section). Moreover, URP can make use of this information in the context of particular functional forms: both for consistency checking and for deriving constraints on parameters.

Some qualitative properties uniquely determine an analytical form for the utility function. Suppose we know something about the risk attitude of the patient in this case, namely that

(constant-risk-posture *life-years***)**

holds.[2] From this, URP concludes

$$u(x) = ae^{-cx} + b \tag{2.1}$$

An analysis of this function (with the help of Sacks's QM [89]) reveals that it is only increasing if a and c are of opposite sign. This fact is recorded, and may be employed in further inference.

Asserting the qualitative behavior **risk-averse** is equivalent to saying that the utility function is concave. The function u (equation 2.1) is concave if and only if $a < 0$; from the monotonicity assertion above URP can also deduce that c must be positive.

If instead **risk-neutral** holds, URP determines that the utility function must be $u(x) = ax + b$, with $a > 0$ if preferences are increasing. Usually we would also scale u from zero to one, giving the equalities $u(0) = 0$ and $u(1000) = 1$.[3] Using this, URP can solve for a and b, giving

$$u(x) = 10^{-3}x \tag{2.2}$$

Note that this function implies a strategy equivalent to the "maximization of life expectancy" criterion used in the original decision analysis for this case.[4] Using URP, the risk neutrality assumption is explicit and easily retractable.

---

[2]To launch into a discussion of the utility-theoretic principles of risk attitude would take us too far away from the point at hand. For now it should suffice to take **constant-risk-posture** as a generic qualitative property that it might make sense to assert (or for the program to infer from other information). The purpose of this chapter is to convey a general feel for the types of reasoning performed by the program. A deeper discussion of the utility theory involved is reserved for chapter 3.

[3]In this example we have arbitrarily set the upper bound for *life-years* to 1000 (measured in hundredths of a year). Note that we only know that this is the "best" value because **monotonic-increasing** was asserted.

[4]If f is a utility function for x, any function g such that $g(x) = af(x) + b, a > 0$, has the property

Reasoning about qualitative behaviors of utility functions can also work in the opposite direction. We can assert that a particular functional form holds (perhaps based on some well-studied utility model) and ask URP about its qualitative behavior. Figure 2.1 illustrates an example terminal interaction (user input follows the "-->" prompts, my comments are in italics) where the user asserts a utility function for *life-years* of the form $u(x) = (x+b)^{-c}$, with b positive and $-1 < c < 0$.[5] Here URP was able to determine that preference is increasing and decreasingly risk averse. Without the constraint on c, URP can only determine that the function is monotonic (in one direction or the other).

The manipulations of qualitative behaviors I have been presenting in this section do not really have much to do with the particular medical example we are discussing. Rather, the exercise is meant as an illustration of what qualitative behaviors look like and a demonstration of some of the things we can do with them in URP. They can be extremely useful for decision making when they determine a functional form (as **risk-neutral** and **constant-risk-posture** do above), or constrain a pre-existing functional form. A further use of qualitative behaviors for deriving decisions is illustrated in the next section, on dominance testing. Discussion of the general value of reasoning based on qualitative assumptions pervades this thesis, particularly the conclusion.

2.3 Dominance Testing

The ultimate goal of the preference modeling task is to show that certain strategies are preferred to others. Since our incomplete preference models do not admit the calculation of unique preference measures, we must resort to more general *dominance-testing* routines to prove that a particular strategy is best for all admissible utility functions. The general dominance problem is discussed in section 4.1.

Recall from above that when **risk-neutral** is assumed (as in the Tufts consult) the function u is completely determined. In such a situation, URP would choose the

that $f(x) > f(y) \Leftrightarrow g(x) > g(y)$. In this case we say that f and g are *strategically equivalent* utility functions. Given that utilities are finitely bounded, it is always possible to generate a utility function strategically equivalent to f having any finite range. Thus, arbitrary scaling from zero to one is allowed, and it was really not necessary to solve for a and b in this case. When using u as part of a multiattribute function, however, these constants will matter.

[5]Naturally, only users experienced in utility analysis would use URP in this fashion. More realistically, these forms would come from some other program using URP and would be invisible to the user.

```
command --> ASSERT-FUNC-FORM
attribute --> life-years
Start of interval --> 0    end of interval --> 1000
```

All behavior and functional form assertions may be specified over arbitrary intervals of the utility function. Here I chose an upper bound of 1000 (hundredths of a year), a reasonable maximum for this patient.

```
Enter the functional form for the utility function
   --> (expt (+ x b) (- c))
Setting functional form for attribute a to:
     1
  --------
         C
  (X + B)
```

Functional form specifications may be built up from the standard operators, written in prefix notation. The form is displayed in a more reasonable form to catch typographical errors.

```
  Enter constraint --> (> b 0)
  Enter constraint --> (< c 0)
  Enter constraint --> (> c -1)
  Enter constraint --> ()
```

Before analysis, the user enters constraints on the parameters of the function. Finally, we ask URP *to determine which of the qualitative behaviors it knows about apply to this function.*

```
command --> DESCRIBE-BEHAVIORS
attribute --> life-years

 MONOTONIC-INCREASING holds
 RISK-AVERSE holds
 DECREASING-RISK-AVERSE holds
 INCREASING-PROPORTIONAL-RISK-POSTURE holds
```

Figure 2.1: Determining the qualitative behavior of an asserted functional form

option which maximized equation 2.2, in this case the strategy **both**. Of course, if we always had such strong constraints we would not need a program with the generality of URP.

In the more common (and more interesting) case, we will have only a constrained analytical form, or perhaps not even that. Fortunately, there are often conclusions we can reach with only very weak qualitative restrictions on the utility function. Researchers in the field of *stochastic dominance* [112] have developed a collection of criteria under which prospects can be shown to be ordered for certain classes of utility functions. Each *dominance algorithm* is associated with a well-defined class of utility functions U. When run on the prospects p_1 and p_2, the algorithm returns **true** only if $p_1 \succ p_2$ for all $u \in U$. A deeper discussion of stochastic dominance and how it is used in URP appears in chapter 4.

For example *first-order dominance*[6] is associated with the class of monotonically increasing utility functions. Therefore, whenever we know that preference for an attribute is monotonically increasing, we can use the first-order dominance algorithm to partially order prospects over that attribute. *Second-order dominance* applies to the concave (**risk-averse**) utility functions.

Figure 2.2 depicts the results of applying the first- and second-order dominance algorithms on the prospects associated with the strategies in our medical example. In the diagram, there is a single arrow from strategy A to strategy B if and only if A first-order-dominates B. Double arrows represent second-order dominance. As we can see, several of these strategies considered may be eliminated with relatively weak restrictions on the utility function for *life-years*.

It is interesting to note in figure 2.2 that four of the seven options are ruled out under the trivial assumption that more *life-years* are preferred to less. The second-order links are redundant in this example, but they are still useful to record just in case some of the options are unavailable for some reason. These relations hold as long as the patient is not risk prone for length of life—usually considered a very reasonable assumption. An in-depth discussion of why a person may or may not be risk averse for *life-years* is given in section 8.4.2 below. Note that the only undominated options are **cabg**, **both**, and **no-repair**. The first two are not surprising, since they are the strategies recommended by the original consult. And even though **no-repair** is by far the worst strategy with regard to expected lifetime, the fact that it is undominated makes sense. It is the only strategy that does not include an immediate surgical procedure, and therefore the only one that

[6] Fishburn and Vickson [38] describe this as well as all other dominance conditions mentioned here.

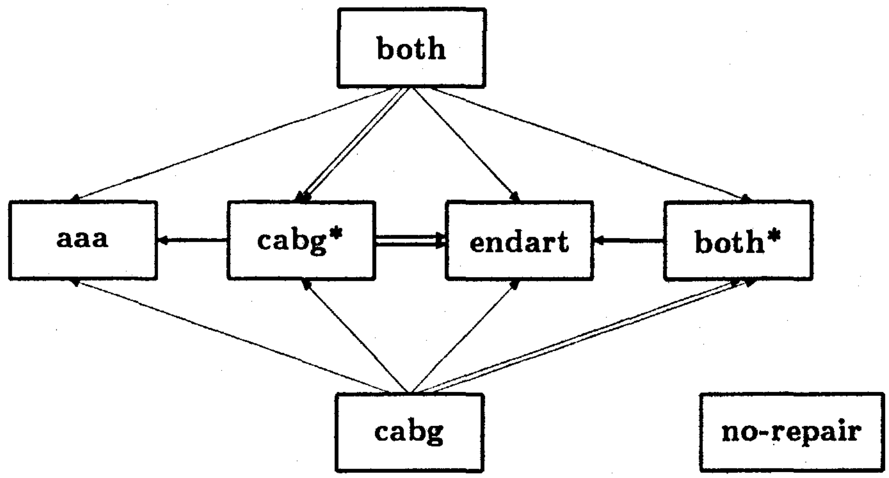


Figure 2.2: Dominance relations in the medical example

does not present a significant risk of immediate death. A utility function that placed a large value on life in the very short term (perhaps an extremely risk averse form) would rank **no-repair** the highest—such utility functions may be consistent with the conditions of first- and second-order dominance. In this case, establishing that the patient is not *extremely* risk averse (a relatively easy task from the perspective of assessment) should eliminate **no-repair** as a viable option.

In other cases, it may turn out that first- and second-order dominance conditions are not sufficient to rank the competing strategies. Such a result would indicate that stronger assumptions about the preference model are required. Note that application of a particular functional form (such as $u(x) = x$ used in the consult) would not uncover this kind of sensitivity in the model.

2.4 Multiattribute Utility

In this section I explore the possibilities of extending the analysis of this case to include other factors in addition to *life-years*. Consideration of outcomes with more than one dimension requires the construction of a *multiattribute* utility function. URP has considerable facilities for structuring multiattribute models, however there are no implemented procedures for mathematical reasoning about these models. Though multiattribute utility is not useful for determining decisions in URP, expertise in structuring these models is useful and interesting in its own right.

It is easy to imagine decision-making criteria other than length-of-life that may be important to consider in this case. I have already mentioned that there is a possibility that the patient will suffer a disabling stroke; he may not be indifferent between this outcome and immediate death. The strategies include major surgical procedures which can have a significant impact on the patient's quality of life, in addition to their influence on survival. Finally, financial cost may enter into the decision, as the various strategies differ greatly in that dimension.

The attributes we will consider for this example, then, are length-of-life, whether or not the patient suffers a disabling stroke, the morbidity (pain and discomfort) associated with the major surgical procedures (CABG and carotid endarterectomy), and financial cost. For the remainder of this discussion I will use the attribute names and shortened notation given in table 2.2.

$$\begin{aligned} X_1 &\equiv \textit{life-years} \\ X_2 &\equiv \textit{disabling-stroke?} \\ X_3 &\equiv \textit{endart-morbidity} \\ X_4 &\equiv \textit{cabg-morbidity} \\ X_5 &\equiv \textit{cost} \end{aligned}$$

Table 2.2: Notation for utility attributes in the medical decision example

The rest of this discussion is concerned with finding a form for the multiattribute utility function $u(x_1, \ldots, x_5)$. Finding a form consists of decomposing the multivariate function into combinations of functions of smaller dimension, usually culminating in utility functions of a single attribute. URP contains a substantial knowledge base of specifications for these decompositions, and theorems that prescribe the conditions under which they are valid. These conditions are usually expressed as independence axioms which constrain the way in which preferences over subsets of the attributes may interact. Independence axioms, theorems, and decomposition in general are discussed at length in chapters 5 and 6.

The following example illustrates one possible way in which URP's knowledge about independence axioms may be used for this medical decision situation. Suppose that we are willing to make some independence judgments[7] about preference

[7] Once again, defining what these particular independence concepts mean would take us too far afield at this point. For now I am just trying to convey a feel for the kinds of reasoning URP performs in multiattribute decomposition.

over the attributes $X_1, \ldots, X_5$. In particular, we assert:

1. (**utility-independence** $(X_2\ X_3\ X_4\ X_5)\ (X_1)$)
2. (**utility-independence** $(X_1\ X_3\ X_4\ X_5)\ (X_2)$)
3. (**utility-independence** $(X_1\ X_2\ X_4\ X_5)\ (X_3)$)
4. (**utility-independence** $(X_1\ X_2\ X_3\ X_5)\ (X_4)$)
5. (**utility-independence** $(X_1\ X_2\ X_3\ X_4)\ (X_5)$)

URP can quickly determine that a valid decomposition structure is:

1. (**multiplicative-form** $(X_1\ X_2\ X_3\ X_4\ X_5)$)

Assumptions 1 through 5 above say that preferences over prospects involving any four of the attributes do not depend on the fixed value of the fifth. These are fairly strong assumptions, and indeed, the multiplicative utility function is one of the simpler multiattribute forms known to URP. Decomposition conclusions such as this can be graphically displayed in the form of figure 2.3.

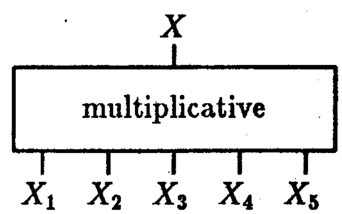


Figure 2.3: X decomposed via the multiplicative functional form

Now, suppose we decide that some of the original assumptions do not hold after all, because relative preference for outcomes containing $X_3, \ldots, X_5$ depends crucially on length of life and the presence or absence of stroke. For example, the patient might be more willing to trade possible short-term morbidities (attributes X_3 and X_4) for a reduced chance of disabling stroke (X_2) if he is going to live longer. Similarly, tradeoffs between morbidities and length of life may be different for patients who have suffered disabling strokes. Therefore, we must retract assumptions 1 and 2. Since all URP assumptions are maintained as propositions in an underlying truth maintenance system (part of McAllester's *R*easoning *U*tility *P*ackage [73], or RUP) such modular assertion and retraction is easy to perform. URP determines that the multiplicative form is no longer valid for X (retracting conclusion 1).

Perhaps we are willing to make additional assumptions in lieu of 1 and 2. In particular, suppose we assert that preference for morbidities and costs are invariant when both X_1 and X_2 are fixed, and that preference for lotteries containing X_1, X_4, and X_5 does not depend on the other attributes.

6. **(utility-independence** $(X_3\ X_4\ X_5)\ (X_1\ X_2)$**)**
7. **(utility-independence** $(X_1\ X_4\ X_5)\ (X_2\ X_3)$**)**

With these replacement assumptions, the multilinear form is valid, as are several other propositions, including conclusion 3.

2. **(multilinear-form** $(X_1\ X_2\ X_3\ X_4\ X_5)$**)**
3. **(utility-independence** $(X_4\ X_5)\ (X_1\ X_2\ X_3)$**)**

Another alternative is to decompose X hierarchically, combining the attributes X_1, X_2, and X_3 into a single vector attribute, W_0. Attributes X_4 and X_5 are similarly combined to form the vector W_1. The utility independence of W_1 from W_0 (conclusion 3) directly validates a two-attribute functional form which I will refer to as the "UI form." To complete the decomposition we need to find a multiattribute form for the attributes composing W_0 and W_1. This is accomplished by applying the usual procedure recursively to a subset of the original universe of attributes. In this case, URP is able to determine that a multiplicative utility function is valid for W_0 and that the UI form is valid for W_1.

4. **(multiplicative-form** $(X_1\ X_2\ X_3)$**)**
5. **(utility-independence** $(X_4)\ (X_5)$**)**

The resulting decompositions are depicted in figure 2.4.

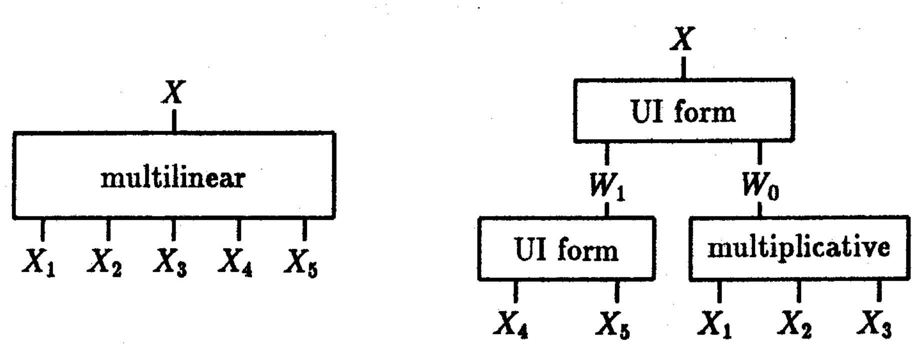


Figure 2.4: Two possible decompositions for X based on assumptions 3 through 7

In this case, there exists a simple single-level decomposition using the multilinear form and a two-level decomposition requiring intermediate vector attributes (introduced automatically by URP). But while the hierarchy for the decomposition on the left-hand side of figure 2.4 is simpler, the utility function corresponding to the multilinear form for X contains 30 $(2^5 - 2)$ independent scaling constants compared to only seven (three for the multiplicative decomposition of W_0 and two each for the UI forms) for the decomposition on the right. Based on this simple metric, the second decomposition would be chosen for the preference model.

Similarly, we could retract any of the other assumptions, perhaps replacing them with weaker or stronger independence conditions described in URP's knowledge base. Each time a replacement is made URP restructures the preference model according to which utility functions are justified by the assumptions. Through automatic hierarchical decomposition, the program is able to construct models ranging over a continuum of possible structures.

3. Reasoning About Qualitative Behaviors

Utility theorists have developed a large vocabulary for describing general properties of utility functions that helps bridge the gap between intuition and mathematics. A major goal in the URP project has been to show that it is possible to perform the greatest bulk of reasoning about preferences in terms of these properties, and that conclusions based on them will be more satisfying than those derived under a purely quantitative approach. Qualitative behaviors of single-attribute utility functions make up one very important category of qualitative property, but they are by no means the only type. This chapter describes URP's facilities for reasoning about them, with special emphasis on mechanisms necessary to produce the example of section 2.2. Reasoning about independence axioms—the second major type of qualitative property used by URP—is the subject of chapter 5.

3.1 Functions in URP

Recall that an URP preference model consists of a collection of assertions about a multiattribute utility function. Multiattribute utility functions themselves are often defined in terms of other utility functions, usually culminating in single-attribute functions. In reasoning about overall preferences, we consider various properties of the unidimensional functions as well as the possible decomposition patterns. It is fairly obvious that functions are a central object of all of this reasoning, and that a large fraction of assertions about utility structure can be expressed in terms of properties of particular functions making up that structure.

For the remainder of this discussion I will use the term *function* to mean the object in URP that *specifies* a function of a single numeric variable. Since these specifications are incomplete, we cannot in general manipulate them as we could explicit mathematical functions. Nonetheless, there is a substantial variety of kinds of reasoning we can perform with them.

While URP employs functions other than utility functions (such as risk functions, to be discussed below), these special functions are the main objects of concern. A utility function u is a mapping from Y to $[0,1]$, where $y \in Y$ describes some part of the outcome space. More formally, $Y = Y_1 \times \cdots \times Y_m$, where $\{Y_1, \ldots, Y_m\}$ is a subset of $\{X_1, \ldots, X_n\}$. In the unidimensional case (the one we are most concerned with here), y will be the value of a single attribute, and we will say $u_i : X_i \to [0, 1]$.

Two components of a function are therefore its domain and range. There are always two distinguished points in the domain, the most and least preferred, denoted by y^* and y_*, respectively. These may or may not be known, or there may be constraints on the possibilities. There may be other outcomes which are indifferent (equally preferred) to y^* or y_*.

Three other function components restrict the possible mappings between the domain and range: *functional forms*, *evaluations*, and *qualitative behaviors*. Each of these is described in greater detail below. Representation for URP functions is based on the QM package developed by Sacks [89]. Much of the mathematical reasoning about functions is performed by QM, described in section 3.3.

3.1.1 Functional Forms

The *functional form* is an expression which describes the computation of the function's value from the input. This description is built from standard arithmetic operations and conditional expressions, over numbers and symbolic parameters. Symbolic parameters may be associated with arbitrary constraints or relations to other URP objects.

Functions that have different symbolic forms over different slices of their domain are represented by piecewise function intervals. Piecewise functions fit in naturally with QM's analysis of functions as collections of monotonic intervals.

3.1.2 Evaluations

Evaluations are objects that represent the value of the function at specific points. Like symbolic parameters, these objects may be defined by arbitrary relations to other objects in the system. A function may include any number of these evaluations. Evaluations typically become useful in reasoning about utility assertions in the form of preference choices. Utility functions all include the two special evaluations $u(y_*) = 0$ and $u(y^*) = 1$.

3.1.3 Qualitative Behaviors

Qualitative behaviors[1] are overall constraints on a function's characteristics that can be described by a single proposition. Part of the value of these constraints is that they restrict the possibilities for the other components of URP functions: functional forms and evaluations, as well as the subcomponents making up these objects. The qualitative behaviors are treated as first-class components because they may have significant interactions (that is, certain combinations of behaviors may have special interpretations) and because their meaning may vary for different functional forms. Some qualitative properties (including **monotonic-increasing**) were illustrated in the example of section 2.2. A list of the qualitative properties of unidimensional utility functions currently represented in URP is given in appendix A.

Most of these qualitative behaviors only have meaning if the domain of the function is a set of numeric values, such as subranges of reals or integers. If Y is made up of non-numeric values (such as vectors or discrete categories) URP may apply a mapping from the members of Y to some numeric domain. Discrete categories, for example, would map to an index set of integers.[2] Then the program can use the same qualitative propositions to describe the index set, taking care to make the correct type conversions when making inferences about the utility function from them. These issues are discussed more fully in section 8.3.1.

3.2 Manipulating Qualitative Behaviors

3.2.1 Qualitative Behavior Assertions

Like assertions about independence axioms, qualitative behavior assertions are handled by RUP. A qualitative behavior assertion is a two-place predicate, where the arguments indicate the utility attribute and an interval of values over which the behavior applies (the interval is omitted in the notation below when the behavior holds over the entire domain of the attribute). Logical relations among the qualitative behaviors are specified and maintained by RUP. Inference across intervals

[1] Behaviors of functions are a specific type of property that may be expressed qualitatively in URP. When it is clear that we are talking about functions, the terms *qualitative properties* and *qualitative behaviors* may be used interchangeably.

[2] It may even be reasonable for URP to change this mapping dynamically during preference modeling. For example, it will usually be most useful to map the categories in a way that preserves preference order—which may not be known at the outset.

is handled specially, for reasons of efficiency. Some illustrative logical relations are given in figure 3.1.

(risk-neutral ?f ?i**)** $\Leftrightarrow$ **(constant-proportional-risk-posture** ?f ?i**)** $\wedge$ **(constant-risk-posture** ?f ?i**)**

(monotonic ?f ?i**)** $\Leftrightarrow$
(monotonic-nonincreasing ?f ?i**)** $\vee$ **(monotonic-nondecreasing** ?f ?i**)**

(decreasing-risk-aversion ?f ?i**)** $\Leftrightarrow$
(decreasing-risk-posture ?f ?i**)** $\wedge$ **(risk-averse** ?f ?i**)**

Figure 3.1: Logical relations among qualitative behaviors of functions

In URP, a behavior that is fully characterized by a combination of lower-level behaviors is treated as a terminological definition, resulting in simpler specification of the logical relations and mathematical properties of the behavior.

3.2.2 Qualitative Behaviors and Function Components

The mechanisms by which qualitative behaviors affect other function components are a bit more complex than the logical relations above. Qualitative behaviors can determine the functional form, they can constrain parameters of a particular form, or they can restrict the value of evaluations. There are two alternate implementation approaches; URP employs a combination of these.

Surface-Level Approach

The simplest approach is to associate each significant combination of qualitative behaviors with its implications for the functional form. For example, the knowledge base of a purely surface-level reasoner would contain many logical statements looking something like

(risk-neutral ?f**)** $\Leftrightarrow$ *functional-form*(?f) = **linear-form**

where the structure of **linear-form** is specified elsewhere as $ax+b$. Also associated with **linear-form** would be the implications of various other qualitative properties on the parameters. For example, if $u(x) = ax + b$, then

(monotonic-increasing u**)** $\Leftrightarrow a > 0$.

These sorts of inference are easily handled by the TMS within RUP. The implications that result in a functional form are part of the original knowledge base, instantiated at the same time as the function object. The implications from functional forms and further qualitative behaviors to parametric constraints are implemented as pattern-directed inference rules, activated only when the functional form is established. These noticers also contain LISP code to directly modify the data structures representing the function object.

A Deeper Mathematical Approach

The problem with the above approach is that it does not capture the deeper mathematical meaning of the qualitative behavior assertions. Because of this, it is necessary to anticipate the effect of every qualitative behavior on every different functional form and combination of qualitatively significant parametric constraints.

For example, consider the interaction between the useful qualitative behavior **constant-risk-averse** and the sign of the slope of a monotonic utility function, u. Reasoning at the surface level, the knowledge base would have to contain separate rules to determine the functional form given **constant-risk-averse** and each of the two direction possibilities. In this case, the two rules would be:

1. (**monotonic-increasing** u) $\Leftrightarrow u(x) = ae^{-cx} + b, a < 0, c > 0$
2. (**monotonic-decreasing** u) $\Leftrightarrow u(x) = ae^{-cx} + b, a < 0, c < 0$

As shown, the two functions differ only in the sign of the exponent. Similar symmetries will be explicitly represented in describing the interactions of the two types of monotonicity with other qualitative behaviors (**risk-neutral** and **constant-risk-prone**, for example). Other natural symmetries (or other simple structural relationships) are also ignored by the surface-reasoning system, such as that between risk aversion and proneness, increasing and decreasing risk postures, etc.

This duplication is particularly irritating because we are failing to take advantage of the very elegant structure of the concepts underlying the qualitative behaviors useful in utility theory.[3] The alternative is to represent these qualitative behaviors in terms of their mathematical interpretations. These concepts generally have a strict mathematical definition which is largely independent of the other qualitative behaviors that apply in a particular case.

[3]This structure is apparent on a simple perusal of the list of qualitative behaviors given in appendix A.

Let us look at risk aversion, since most of the unidimensional qualitative behaviors defined thus far deal with that phenomenon. The risk aversion function r associated with a monotonically increasing utility function u is defined as

$$r(x) = \frac{-u''(x)}{u'(x)} \tag{3.1}$$

For monotonically decreasing utility functions, the risk aversion function is called q, with

$$q(x) = \frac{u''(x)}{u'(x)} \tag{3.2}$$

For the rest of this discussion, all references to r are meant to indicate q for decreasing functions (non-monotonic functions are ignored for the time being).

With this definition, risk aversion has a simple meaning. The assertion (**risk-averse** u) is now equivalent to saying that for all x, $r(x)$ is positive.[4] Given the relationship between r and u, this piece of information is sufficient for all inference that was possible in reasoning about risk aversion on the surface level. More importantly, representing the assertion at this level provides for greater orthogonality between qualitative behaviors. Now, for example, the effect of asserting (**constant-risk-posture** u) is simply to add the fact that $r(x) = c$, for some constant c. Taken together, it is quickly seen that $c > 0$.

Not quite as quickly, it would also be possible for the system to derive the functional form directly from these mathematical descriptions of the qualitative behaviors. Substituting the definition of r, we get a differential equation

$$u''(x) + cu'(x) = 0$$

which can be solved to determine the functional form of u (here $ae^{-cx} + b$, with $a < 0$ and $c > 0$, as above). So we see that it may not even be necessary to explicitly include relations from qualitative behaviors to functional forms at the surface level at all.

Of course, this is how utility theorists determined these functional forms in the first place. The results, usually along with the derivations, appear in textbooks. But it also seems wrong to have to perform this derivation each time we encounter this combination of qualitative behaviors. After all, the notion of constant risk

[4] Actually, for this example, all we really need to know is that u is concave (that is, the second derivative is negative). Thus, we may consider risk aversion for even nonmonotonic functions. For higher-order risk properties it will be necessary to reason in terms of the risk function r.

aversion was deemed important enough to give it its own name, so there must be some reason to think of it at that surface level. Even though utility analysts are mathematically oriented, my bet is that they would prefer to memorize functional forms or look them up in reference books than to derive them from first principles each time.

Having come full circle, the answer to the choice of approaches seems clearer. For useful combinations of qualitative behaviors, it is worthwhile as well as legitimate to enumerate their effects on functional forms and parameters directly. However, because URP may encounter previously unanticipated combinations of these behaviors, it is necessary to develop a deeper mathematical representation for their meaning. As QM has proven powerful enough to determine parameter constraints from qualitative behaviors, and vice-versa, URP uses it for deep reasoning about parameters. Functional forms are generated from surface-level relationships.

3.3 Qualitative Mathematics

URP's mechanisms for reasoning about the qualitative mathematical behavior of functions are based on the Qualitative Mathematics package (QM) developed by Sacks [89]. QM uses calculus and other mathematical techniques to analyze the behavior of piecewise continuous functions of real numbers. While designed primarily for reasoning about the dynamic behavior of physical systems, QM's representations and capabilities for manipulation of function objects have proved quite useful for analyzing the behavior of utility functions. This section briefly describes some aspects of QM, and how some of the program's mechanisms have been adapted for use in URP.

3.3.1 Functions in QM

QM represents a function as a collection of intervals on which it is continuous and monotonic. Each interval (called a *fun-int*) is a data structure which records such information as the function's form, inverse, derivatives, and boundary points. QM constructs these representations from a closed-form expression made up of symbolic parameters (possibly constrained), standard arithmetic operations, and a variable. Since a function may have different qualitative behaviors for different values of its parameters, QM generates separate function objects for each distinct case, recording the parameter ranges for which it is appropriate. The algorithms and data structures are described in full detail in Sacks's original thesis report [89].

3.3.2 Partially Specified Functions

In applying QM to the analysis of utility functions in URP, it was necessary to extend the function object representation to handle a broader range of incomplete specifications. We often have quite a lot of information about an URP utility function (qualitative behaviors, constraints on evaluations) without an analytic expression. Existing QM data structures can accommodate the expression of this partial knowledge with only slight augmentation. For example, it was necessary to add a facility for specification of information about evaluations, and for expressing disjunctions of possible directions for the function on an interval.

3.3.3 Encoding Function Behaviors

For reasoning about qualitative behaviors as in section 3.2.2, we have to relate the behaviors to mathematical properties represented by QM functions. Fortunately, many of the behaviors of interest for utility functions are directly included in the function object data structure, or are easily computed from it. In particular, direction and convexity have built-in predicates in QM.

Given an incompletely specified function and a behavior, we would like to know whether the function exhibits that behavior, and what implications an assertion that the behavior holds would have on the function specification. Each URP qualitative behavior is encoded as a set of three procedures, operating on a function object and an interval (range of domain values):

1. A possibility predicate (*pos*), which is true if the behavior *possibly* holds over the indicated interval of the function

2. A necessity predicate (*nec*), which is true if the behavior *necessarily* holds over the function interval

3. An enforcer procedure (*enf*), which constrains the function object to observe the behavior over the interval

These procedures are encoded as (generally short) fragments of LISP code. As mentioned above, some behaviors are completely defined in terms of other behaviors, and therefore do not require their own procedure specifications.

Behavior	Form name	Expression
risk-neutral	linear	$ax + b$
constant-risk-posture	exponential	$ae^{-cx} + b$
constant-proportional-risk-posture	power	$ax^{1-c} + b$
(two possibilities)	log	$a(\log x) + b$

Table 3.1: Qualitative behaviors that imply single-attribute functional forms

For example, the *nec* predicate for **risk-averse** simply checks that all QM *fun-ints* over the interval of interest are marked *concave*. Similarly, the *pos* predicate checks that none are *convex*. The *enf* procedure marks as concave any *fun-int* of unknown convexity. Some of the *enf* procedures corresponding to other behaviors are significantly more complicated.

An URP function object in general includes a list of possible QM functions representing it. This disjunctive form allows for conditioning dependent on the value of parameters of the analytical form. Because the URP behavior procedures described above work for individual QM function objects, a predicate over the entire URP function is a conjunction of *nec* predicates or a disjunction of *poss*. Enforcing a behavior over an interval involves discarding any of the possible QM functions that is inconsistent with the behavior (that is, not *pos*), and applying the *enf* procedure to each one remaining.

Thus, one way that constraints on parameters of symbolic forms are determined is by elimination of cases generated by QM. For the types of behaviors implemented and the functional forms analyzed in tests of the program, QM generates all of the necessary distinguishing cases. We should expect this to work as long as the qualitative behaviors that are important for utility correspond to qualitative properties of functions known to QM. In URP, all of the behaviors are first-class QM concepts of the utility function or of functions derived from the utility function.

3.3.4 Enforcing a Functional Form

The behaviors in URP that determine a symbolic form for the utility function are particularly useful. There are three of these, shown with their associated forms in table 3.1.

When one of these behaviors is asserted, QM creates a list of possible function objects representing the function (differing in parameter ranges). URP merges these objects with the current incomplete description of the utility function, discarding the QM possibilities which are inconsistent with what is already known. For the forms appearing in the table, direction and convexity information constrains the values (usually the sign) of parameters a and c.

3.3.5 Risk Functions

To reason about the risk properties of utility functions (which most of the defined behaviors describe), URP analyzes the risk functions defined in equations 3.1 and 3.2. QM is used to generate a function object corresponding to the ratio of the second and first derivatives, created from symbolic expressions already computed by QM. Thus, behaviors dealing with high-order risk properties correspond to qualitative properties of the risk function r. For example, **increasing-risk-averse** holds if and only if r is positive (or equivalently, u is concave) and increasing.

Similarly, behaviors dealing with proportional risk posture are treated as qualitative properties of the function $\rho(x) = xr(x)$.

3.3.6 Describing Behaviors

Using QM in conjunction with the *pos*, *nec*, and *enf* mechanism, it is possible to assert and analyze a wide range of combinations of qualitative behaviors, functional forms, and parametric constraints. To test this capability, I tried the system on all of the single-attribute utility functions appearing as examples and exercises in chapter 4 of Keeney and Raiffa [61]. Of these 18 diverse functions, URP was able to generate correct behavior descriptions of the sort depicted in figure 2.1 for all but one.[5] In addition to testing for behaviors, the system may be used to derive constraints on parameters for all of these functions, or to display the risk or proportional risk functions associated with them.

3.4 Qualitative ⇒ Intuitive?

The preceding discussion illustrated some of the qualitative properties URP knows about and how they influence the mathematical utility model. An assump-

[5] This exception only failed in the analysis of the proportional risk behavior, due to a fairly complex expression for the proportional risk function. As a further note of interest, the system uncovered a typographical (sign) error in table 4.5 of the aforementioned text.

tion that has been implicit throughout is that reasoning in terms of qualitative behaviors is somehow more intuitive than the usual numeric approach. The reader may reasonably protest, however, that the above properties are not very intuitive (except perhaps to utility experts) and therefore not necessarily helpful. I would contend, though, that they are only one step away from being intuitive. That is, each property (risk aversion, proportional risk aversion, etc.) will have a more intuitive interpretation in the context of a particular attribute or decision domain. URP uses the technical terminology because it is more general and more concise. Nevertheless, I will concede that a system that cannot tie utility-theoretic concepts to domain principles will not be very useful to a decision maker unfamiliar with utility theory. This issue will be addressed at length below in chapter 8, about preference knowledge for specific application domains.

4. Stochastic Dominance

Although the qualitative analysis of utility function behaviors is interesting in its own right for the insight it provides into the structure of preferences, we are at least as interested in the implications these qualitative behaviors have for particular decision situations. As we have seen, URP can represent a wide range of partial specifications for utility functions, incorporating constraints from different sources into a central function object. The use of this function object as a basis for prescribing decisions raises some interesting questions. In this chapter I discuss the general problem, outline some approaches, and describe the mechanisms employed by URP. Possibilities for extending URP's capabilities in this area are also discussed.

4.1 Dominance

4.1.1 The General Case

Testing for the dominance of prospect p_1 over prospect p_2 is a search for a proof that $u(p_1) > u(p_2)$ given all the known constraints on u. Taking the definition of expected utility, this is equivalent to

$$\sum_{x \in X} p_1(x)u(x) > \sum_{x \in X} p_2(x)u(x) \tag{4.1}$$

where $p_i(x)$ is the probability of outcome x in prospect p_i. In the most general case, u is specified to an arbitrary level of completeness and the search for a dominance proof is arbitrarily difficult. Therefore, the basic strategy is to avoid the general case by recognizing features of u which indicate the relevance of special-purpose strategies. But since a wide range of different special-purpose strategies may be employed depending on the features of u, the overall system approaches generality as more feature-strategy combinations are incorporated.

4.1.2 The Case of Known Functional Forms

An enormous improvement from the general case is obtained when we have a symbolic form for the utility function. In this situation the inequality 4.1 is a closed-form expression, annotated by any known constraints on the parameters of u. Overall behavior information or constraints on evaluations of u at specific points may also be known, but these are harder to use if they are not translated to parametric constraints. Though URP is often able to generate these constraints from qualitative behaviors (as described in section 3.3 above), there are many situations where it cannot, or where the information is simply not expressible in this form.

Even when functional forms are known, dominance-proving is a hard problem—one that has long concerned mathematicians, operations researchers, and those developing symbolic algebraic reasoners, among others.[1] Its difficulty will depend on the structure of the inequality expression we are trying to prove. Since each side of 4.1 is a linear combination of utility function evaluations, the expression's complexity is totally determined by the complexity of u. Thus, examinations of u's form will indicate which constraint algorithms will be tractable and correct, or which may be most likely to give us a good answer. Choosing these algorithms and incorporating them into URP is an important area for future research. As we will see in section 7.3, similar problems arise in reasoning about hypothetical preference choices for assessment.

There are some simple tricks we can employ to make the problem somewhat easier. First, we may perform any positive linear transformation of u that would simplify the dominance expression. This will often eliminate one or two parameters (which are usually only included to allow linear transformations for scaling purposes anyway). Thus, the `linear` form implied by the behavior **risk-neutral** can be seen to order all prospects trivially by expected value.

Second, we may simplify the prospects themselves by removing common components. That is, we generate new prospects $p_{1'}$ and $p_{2'}$, where

$$\begin{aligned} p_{1'}(x) &= \mathrm{MAX}\left[0,\ p_1(x) - p_2(x)\right] \\ p_{2'}(x) &= \mathrm{MAX}\left[0,\ p_2(x) - p_1(x)\right] \end{aligned} \tag{4.2}$$

Of course, this transformation applies regardless of u.

Finally, note that deciding the preference order of certain outcomes is usually considerably easier than proving dominance among uncertain prospects. Typically we rely on monotonicity assertions, or transitivity with respect to known ordered pairs. While this may seem a trivial matter, ordering of certain outcomes is a common operation in testing for dominance of more complex prospects.

[1]The volume by Hardy, Littlewood, and Pólya [45] is an early, comprehensive work devoted entirely to the problem of determining that one mathematical expression dominates another. While their interests are of course much different than ours here, it illustrates the extent of mathematical activity in the subject.

4.2 Stochastic Dominance

In the section above I suggested that the way to structure our dominance testing is to match specific strategies to features of the utility function. The next step, then, is to identify the useful features and develop their associated strategies. Fortunately, there is a substantial body of research in the field of *stochastic dominance* [1] [112] which is devoted to precisely this problem. Though developed primarily with applications to finance and economics in mind, stochastic dominance results apply to utility functions in general. Typically, these results associate algorithms for dominance-testing with well-specified classes of utility functions for which they are valid. The dominance-testing procedures are specified solely in terms of the prospects to be ranked.

Fishburn and Vickson [38] describe four major stochastic dominance results, providing successively stronger dominance-proving power while placing greater restrictions on the class of utility functions U they hold for. Table 4.1 lists these dominance categories along with the properties of the utility functions $u \in U$.

Dominance Type	*Properties of U*
first-order	**monotonic-increasing**
second-order	**monotonic-increasing, risk-averse**
third-order	**monotonic-increasing, risk-averse,** positive third derivative
DARA	**monotonic-increasing, decreasing-risk-averse**

Table 4.1: Four dominance categories with their associated utility function classes

Notice that all of the utility-class specifications—with the exception of third-order dominance[2]—can be completely described in terms of qualitative behaviors already defined in URP. Thus URP is able to determine which dominance algorithms are valid for arbitrarily specified utility functions, and to indicate the implications for u of adding the assumptions necessary for a technique to be applicable. This ca-

[2]This deficiency is not serious; including the third derivative's sign as a new qualitative behavior would be a simple matter.

pability underscores the value of reasoning directly in terms of qualitative behaviors for single-attribute utility.

4.3 Encoding Dominance Routines in URP

A dominance test in the URP knowledge base is an association between an algorithm and a set of applicability criteria. An algorithm—coded directly in a general-purpose programming language—is a predicate. Given prospects p_1 and p_2, the predicate returns **true** if and only if it can determine that p_1 dominates p_2. The applicability criteria define the class of utility functions and the structure of the prospects that the algorithm is designed for.

A sample URP dominance test specification is depicted in figure 4.1. It may be considered for outcomes measured on *interval* scales (explained below in section 8.3.1). The input prospects must be described by *simple* probability distributions. Though these are the only kinds of prospects supported in URP, it would undoubtedly prove useful to describe uncertain outcomes with distributions having analytical (possibly continuous) forms. Several dominance algorithms in the literature are specifically designed for probability distributions of particular parametric classes (normal or log-normal, for example).

```
(def-dominance-test  SECOND-ORDER
  :source              "Fishburn and Vickson"
  :outcome-type        '(interval)
  :prob-dist-type      'simple
  :preconditions       '(:totally-ordered?)
  :utility-conditions  '(monotonic risk-averse)
  :algorithm           'second-order-dom?)
```

Figure 4.1: A dominance test specification in URP

It is helpful to observe that the utility function criteria given in table 4.1 are overly restrictive—the dominance tests are actually applicable under more general circumstances. For example, first-order dominance is applicable whenever our specification for u gives a total ordering on the outcomes (or at least those appearing in the prospects under consideration). Thus, the first-order conditions are satisfied by monotonically decreasing utility functions, and even by some functions that are nonmonotonic but nevertheless order the relevant outcomes.

Before running the second-order dominance algorithm (the LISP function named `second-order-dom?`), URP must verify that the outcomes appearing in p_1 and p_2 are totally ordered in preference by the incompletely specified u. This precondition is necessary because the algorithm applies a "$\succ$" predicate in the course of its processing. In usual descriptions of the dominance criteria (as above), the condition **monotonic-increasing** is required so that the program can substitute the $>$ relation over real numbers for $\succ$. With a flexible coding of the algorithm that employs the preference order relation directly, we avoid this unnecessary requirement.[3]

The utility conditions specify the class of utility functions u must belong to, here the **monotonic** (either direction) and **risk-averse** functions. These conditions may be tested with the URP *nec* predicates associated with the behaviors.

Only a few dominance tests have been specified (and the algorithms coded) for use by URP. These are the first-, second-, and third-order dominance classes defined by Fishburn and Vickson [38]. URP also may try "zero-order" dominance, which holds only if the worst possible outcome in p_1 is preferred to the best in p_2. This test does not even require total ordering, just the ability to pick the worst and best from p_1 and p_2 and to order that pair. Naturally, zero-order dominance holds in only the most trivial of cases. I would argue, though, that it is useful to be able to recognize the degree of triviality of a particular case.

These mechanisms, in conjunction with qualitative behavior assertions, are sufficient to work out the medical decision example of section 2.3. Applying all of URP's dominance tests to the prospects represented by the competing strategies results in the dominance diagram of figure 2.2.

If none of the applicable dominance tests are sufficient to order the prospects, URP must resort to more general methods. In the current state of implementation, this rarely results in a successful dominance proof.

4.4 Multiattribute Dominance

Any discussion of dominance testing in the multiattribute case is necessarily speculative, since URP does not support qualitative behaviors for multiattribute

[3]In generalizing these conditions, it is necessary to make sure that the condition is not otherwise essential to the dominance proof. In this case the criterion **monotonic** is substituted for **monotonic-increasing**. For first-order dominance monotonicity is not necessary at all.

functions, or any mathematical reasoning about such functions. Nevertheless, it is worthwhile to note some of the promising approaches which may be undertaken in a more complete preference modeling system.

Though stochastic dominance research has been mainly concerned with the unidimensional case, recent efforts have extended the techniques to certain multiattribute situations. Since the dominance algorithms are associated with specific forms of the multiattribute decomposition, URP's reasoning facilities regarding multiattribute structure will be very useful for selecting appropriate tests. For example, Mosler [79] reviews dominance conditions for additive decompositions[4] and develops criteria for multiplicative and certain multilinear forms. These latter two forms can be defined in terms of utility independence axioms, which are prominent concepts in URP.

Sarin [91] developed his *evaluation and bound* procedure to take advantage of partial information about the parameters and conditional functions making up a multiattribute utility function (additive or multiplicative). The procedure is particularly useful for deriving bounds on additive forms. He found that this partial information was often sufficient to prove dominance in realistic cases. This finding is consistent with some psychological results (such as the study of Schoemaker and Waid [94]), which suggest that precise numerical weights are often not critical to the predictive validity of linear models.

Recently, Kirkwood and Sarin [64] showed how multiattribute prospects could be ranked with only ordering information regarding the parameters of additive or multiplicative functions. Ordering information is a particularly useful (and quite weak) type of constraint, since it often corresponds to "natural" qualitative behaviors.

I expect that these results could be incorporated into the URP framework in a similar manner to the single-attribute stochastic dominance facilities described above.

[4]The meaning of these terms and URP's facilities for multiattribute decomposition are described below in chapters 5 and 6.

5. Reasoning with Multiattribute Utility Theorems

In this chapter I describe URP's facilities for representing and using knowledge about multiattribute utility decompositions to select a multiattribute utility function. The structure of theorems that can be represented in this framework is defined, and the mechanisms for accomplishing the example of section 2.4 are explained.

These explanations may be excessively detailed at times, particularly in the discussions of theorem interpretation and set-membership constraint propagation (section 5.3). The reader is encouraged to skim over sections which seem to be at a low level of description.

5.1 Representing Independence Axioms

An independence axiom is a relation on attribute sets that restricts the possible interactions of preferences over outcomes described by those attributes. The rich axiomatic structure of multiattribute utility theory is one of its strongest attractions and one of the greatest motivations for this approach. As the fundamental building blocks for reasoning about utility structures, independence axioms play a central role in URP. Representation and manipulation of these objects has been an important focus of URP's design.

5.1.1 Axiom Schemata

Figure 5.1 depicts a typical independence axiom schema. There are 22 axioms known to URP, but not all of these are actually used in theorems. A complete list of these is given in appendix B; descriptions of the axioms appear in the next chapter. The source and definition slots contain information strictly for human eyes. Other slots hold specifications for argument types, logical properties, and relations to other axioms.

The **arg-count** and **arg-relns** slots specify the syntactically legal arguments to the relation. For example, utility independence is a relation of two attribute sets, which must be exclusive and nonempty. This information is used to type-

```
(defaxiom 'utility-independence
    :source ''Keeney & Raiffa p. 284''
    :definition textual definition
    :special-case-of '(preferential-independence
                       generalized-utility-independence
                       interpolation-independence)
    :arg-count 2
    :arg-relns '((exclusive-nonempty-subsets 1 2))
    :rel-prop nil)
```

Figure 5.1: A sample axiom schema

check user assertions, and to constrain the generation of subgoals during theorem interpretation (described below).

The **special-case-of** list enumerates the independence axioms that are strict generalizations of the axiom. Using this, URP can conclude, for example, that

$$(\textbf{utility-independence}\ Y\ Z) \Rightarrow (\textbf{preferential-independence}\ Y\ Z)$$

for any Y and Z. URP only makes such deductions, however, when the more general relation is an active goal. The special-case-of mechanism is much more efficient than using separate theorems for each special case.

The final slot is used to specify any special properties of the relation which may be exploited for more efficient reasoning. Valid **rel-props** include symmetry, transitivity, and other relation properties of that ilk.

5.1.2 Axioms as RUP Propositions

All inferences about independence axioms are performed in RUP. Axiom instances exist as nodes in the truth maintenance system, with the name of the axiom serving as the predicate name. Logical relations between the nodes are set up by URP, either by noticers corresponding to special-case-of conditions and rel-props, or during theorem interpretation.

Assertions and retractions of independence conditions are handled directly by RUP, with certain events in the data base triggering conditions in URP. RUP's dependency-tracing facility is also useful for generating justifications for the program's conclusions.

5.2 Representing Theorems

Multiattribute utility theorems make up a large part of the utility-theoretic knowledge used by URP. These theorems, developed by researchers over the past fifteen years or so, relate combinations of independence axioms to multiattribute utility forms. Many of the theorems can be found in review articles by Farquhar [27] [28] (with a theoretical emphasis), the textbook of Keeney and Raiffa [61] (a more methodological treatment), and separately throughout the literature.

I have developed a representation language for specifying these theorems so that they can be interpreted by URP. There were three principal language design objectives: flexibility, interpretability, and clarity. The first objective refers to the desire to be able to represent a large portion of the available utility theorems. Second, the language should express theorems in a manner easily interpreted by URP. Finally, they should be fairly easy to read by humans. The result is a simple language with a highly constrained logic-like/LISP-ish syntax, which tries as much as possible to state the theorems as they appear in the literature.

An example of an URP theorem is shown in figure 5.2. This theorem is used by the program to derive utility independence relations from others already known to the system. It corresponds to theorem 6.7 of Keeney and Raiffa, and resembles their language quite closely. The rest of the theorems are given in appendix C. Chapter 6 contains descriptions of the content of the theorems.

Here I will only point out some of the main features of the language. A theorem consists of a premise and a consequent, each of which specifies a collection of axioms. The defining expressions may contain variables, either free or bound by universal or existential quantifiers. Variables stand for either attributes or sets of attributes, distinguished by context. Expressions also contain axiom names and constraint terms (union, difference, and overlapping-subset, for example).

In theorem009, X Y1 and Y2 are existentially quantified set variables. The consequent clause consists of the conjunction of five utility independence axioms, which contain no variables not appearing in the premise. URP can make an inference from this theorem if it can find bindings of Y1, Y2, and X which satisfy the premise and the type constraints of the consequents. In that case, it would conclude each of the five consequents computed from those bindings.

The constraint expressions may appear in quantification domains or in arguments to the axioms. An expression consisting of a constraint name and k arguments can be thought of as a $(k+1)$-ary relation holding on the arguments and the

```
(deftheorem 'theorem009
  :source "Keeney & Raiffa p. 316"
  :premise '(for-some-set X
              (for-some-set Y1 in (subset X)
                 (for-some-set Y2 in (overlapping-subset Y1 X)
                    ((utility-independence Y1 (difference X Y1))
                     (utility-independence Y2 (difference X Y2))))))
  :consequent '((utility-independence (union Y1 Y2)
                                      (difference X (union Y1 Y2)))
               (utility-independence (intersection Y1 Y2)
                                      (difference X (intersection Y1 Y2)))
               (utility-independence (sym-difference Y1 Y2)
                                      (difference X
                                               (sym-difference Y1 Y2)))
               (utility-independence (difference Y1 Y2)
                                      (difference X (difference Y1 Y2)))
               (utility-independence (difference Y2 Y1)
                                      (difference X (difference Y2 Y1)))))
```

Figure 5.2: A sample URP multiattribute utility theorem

value of the expression. For quantification domains any relation is usually valid, but for axiom arguments the value must be a function of the expression arguments.[1]

5.3 Goal-directed Inference

The theorem specification language does not restrict the direction of URP's reasoning about the theorems. While a forward-chaining interpretation of the theorems is easiest, when used exclusively it will result in too many inferences which are redundant or irrelevant with respect to the particular problem at hand. The alternative is for the program to reason backward from the goals it is trying to prove. While a backward-chaining approach is prone to the same combinatorial problems, it seems to be more controllable for the types of problems URP is faced with. Therefore, URP contains a fairly sophisticated facility for generating subgoals to pursue when presented with a theorem and a higher-level goal.

[1]This is not strictly required, since typing information implicit in the axiom arguments may be sufficient to uniquely determine the expression's value even if the expression itself is not.

5.3.1 Interpreting Theorems

Interpretation of a theorem in a context consists of reducing the premise and consequent to conjunctions of axiom instances. Such a reduction can be readily converted to a logical implication in RUP. Often, a single interpretation operation in URP will result in several of these reductions.

This reduction is accomplished by binding the theorem variables to the attributes and sets appropriate for the reasoning context, and instantiating the relations appearing in the arguments of the utility axioms. Naturally, the bindings and relations must satisfy the logical structure of the theorem clauses. The theorem interpretation problem can thus be stated succinctly as follows: given a theorem and some binding constraints, find all distinct conjunctions of legal axiom instances that satisfy the logical structure of the theorem. Distinctness is defined in terms of the effect of adding the implication constraint to RUP's database.[2]

The solution cannot be computed directly since the constraint terms do not generally have inverse functions. Exhaustive enumeration of variable bindings is also out of the question since the number of such bindings is exponential in the number of attributes. Therefore, URP has resorted to a fairly complex mechanism based on constraint propagation to generate the variable bindings. The mechanism is fully implemented as described in the next section.

5.3.2 Set-Membership Constraint Propagation

As URP interprets a theorem, each variable it encounters becomes a node in a constraint propagation network. There are two types of nodes, representing sets and elements (in our domain, the elements are always utility attributes). For most of this discussion we will consider only set nodes.

In its least constrained state, a set may be any subset of the attributes. Since we know the n attributes at theorem interpretation time, it would be possible to enumerate these 2^n possibilities. Their number decreases markedly with partial information about which elements are in or outside of the set. To represent this partial information, each node contains an **in** and an **out** list, which hold the elements that are known to be in each of these categories. A set node is maximally constrained when all of the elements in the universe appear in either its **in** or **out** list; this uniquely determines the set it represents. Naturally, an element cannot be on both lists of the same set node.

[2]But we can test for distinctness without using RUP.

Nodes also maintain upper and lower bounds on their cardinality (**max-size** and **min-size**, respectively). Cardinality is restricted directly by the size of the universe and the **in** and **out** lists, and also indirectly through external constraints. For example, axiom arguments are usually required to be nonempty attribute sets. Such external constraints are often useful for determining set membership through elimination reasoning.

Constraints

In the network, set nodes are connected by constraints—objects which enforce some relation among the sets they connect. These relations restrict the combinations of elements each set may contain. The two primitive constraints are **union** and **complement**; all of the other terms appearing in theorems are defined in terms of these. Often, it is necessary to create intermediate set nodes (which do not correspond to any variable) in order to build the more complex constraints. The complex constraints currently defined are given in appendix D.

Let us look, for example, at the **union** constraint. The expression (**union** Y1 Y2) denotes a set Z such that $Z = Y_1 \cup Y_2$. Say that we indeed have three nodes in our network labeled Y1, Y2, and Z, and a **union** constraint connecting them in that manner. The constraint enforces the logical properties given in figure 5.3. The predicates *in* and *out* used in the figure denote the presence of an element on the **in** or **out** list, respectively, and the functions *max-size* and *min-size* give cardinality bounds on the set.

The constraint is enforced by propagation. Whenever a set node adds an element to either the **in** or **out** list, it communicates that fact to each constraint it participates in. The **union** constraint, in turn, checks each of the above relations which may have been affected by the change, and performs any changes to the set nodes necessary to satisfy the condition. Since changes are only made in the direction of further constraint (adding elements to the **in** or **out** lists, tightening the cardinality bounds), the set node operations are determined unambiguously.

The list of properties enforced by the **complement** constraint is easily constructed by analogy to **union**.

This constraint propagation scheme was inspired by the network paradigm of Sussman and Steele [98], and bears a close resemblance to the type of constraint propagation developed by Waltz [108]. The similarity to Waltz is in the meaning of the nodes of the network. A set node really represents a space of possible sets restricted by the **in** and **out** lists and the cardinality bounds. As in Waltz's

For any a:

1. $in(\texttt{Y1}, a) \rightarrow in(\texttt{Z}, a)$
2. $in(\texttt{Y2}, a) \rightarrow in(\texttt{Z}, a)$
3. $out(\texttt{Z}, a) \rightarrow [out(\texttt{Y1}, a) \wedge out(\texttt{Y2}, a)]$
4. $[out(\texttt{Y1}, a) \wedge out(\texttt{Y2}, a)] \rightarrow out(\texttt{Z}, a)$
5. $[in(\texttt{Z}, a) \wedge out(\texttt{Y1}, a)] \rightarrow in(\texttt{Y2}, a)$
6. $[in(\texttt{Z}, a) \wedge out(\texttt{Y2}, a)] \rightarrow in(\texttt{Y1}, a)$
7. $min\text{-}size(\texttt{Z}) \geq \text{MAX}\ [min\text{-}size(\texttt{Y1}), min\text{-}size(\texttt{Y2})]$
8. $max\text{-}size(\texttt{Y1}) \leq max\text{-}size(\texttt{Z})$
9. $max\text{-}size(\texttt{Y2}) \leq max\text{-}size(\texttt{Z})$
10. $max\text{-}size(\texttt{Z}) \leq max\text{-}size(\texttt{Y1}) + max\text{-}size(\texttt{Y2})$

Figure 5.3: Properties enforced by the **union** constraint, $\texttt{Z} = \texttt{Y1} \cup \texttt{Y2}$

line-labeling domain, our combinatorial space is represented compactly, and small incremental restrictions in a single node often reduce the overall size of this space substantially.

The theorem language also provides for variables and constraints that refer to single attributes and relate them to sets. The mechanisms for dealing with these are similar to those dealing with sets, but are more complex. This additional complexity arises because it is necessary to maintain separate contexts corresponding to the possible bindings of universally quantified attribute variables. I will not discuss this mechanism in any detail beyond the simple claim that it works for all of URP's theorems.

Hypothetical Reasoning

As we will see in the example, it is sometimes necessary to make tentative assertions about set membership, due to the incompleteness of our constraint reasoning technique. This incompleteness arises in two ways. The first is the limited expressive power of our representation for sets. With only **in** and **out** lists and cardinality bounds we cannot in general express disjunctions of membership relations. For example, it is not possible to say "$a \in S \vee b \in S$," unless by coincidence this is required by cardinality bounds. Such a condition may be implied by the network (in that assertions that one is *out* will result in the other being placed *in*), but is not observable from the set node in isolation.

The other source of incompleteness is the locality of the constraints. Suppose, for example, we have three **union** constraints:

1. $A \cup B = S_1$
2. $A \cup C = S_1$
3. $B \cup C = S_2$

and are told that the element a is a member of the set S_2 but does not appear in A. A global analysis of the system would result in the conclusion that a must be a member of B, C, and S_1, since all elements not in A must be in both or neither of B and C. Examination of each constraint individually, on the other hand, does not lead to this conclusion.

We can solve this problem with hypothetical reasoning. If we arbitrarily hypothesized in the above example the relation $out(B, a)$, the system would go on to conclude $out(S_1, a)$, $out(C, a)$, and therefore $out(S_2, a)$ from the local constraints. This last conclusion, however, contradicts one of the original specifications. Since all of this reasoning arose from the $out(B, a)$ assumption, we are allowed to withdraw all of these conclusions and assert the assumption's negation, $in(B, a)$, as well as all it implies.

Unfortunately, not all of these difficulties are resolved with single assumptions. In URP's current implementation, the constraint system maintains a stack of hypothesis levels, and backtracks chronologically when contradictions arise. A dependency-directed backtracking mechanism would be much more elegant and efficient, but the expected benefit was not deemed worth the extra implementation effort in the short run. After all, this thesis is supposed to be about preferences, not set membership.

5.3.3 An Example Theorem Interpretation

We are now ready to work through a small example demonstrating theorem interpretation. We will use theorem009, displayed earlier in figure 5.2. For the example, we will only look at the first consequent clause: the one with **(union Y1 Y2)** as its first argument. Suppose our current goal is to conclude

(utility-independence $(X_1\ X_2\ X_3)$ $(X_4\ X_5)$**)**

where $X_1, \ldots, X_5$ are utility attributes (not necessarily the only ones). The interpretation task is to generate all possible bindings for Y1, Y2, and X which satisfy the premise pattern of theorem009.

The interpretation module would first parse the theorem and set up a set-membership constraint propagation network. This network contains set nodes for Y1, Y2, and X, as well some internal nodes created during unification or in decomposing complex constraints. For this example, we need two internal nodes

- $A_1 = \{X_1, X_2, X_3\}$
- $A_2 = \{X_4, X_5\}$

to stand for the arguments to the goal axiom (there will be other internal nodes as well, but these appear at a lower level of description). The network is defined by the following relations:

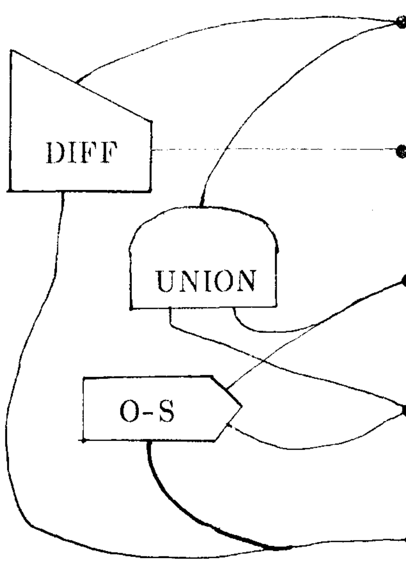
- *union*(Y_1, Y_2, A_1)
- *difference*(X, A_1, A_2)
- *subset*(Y_1, X)
- *overlapping-subset*(Y_2, Y_1, X)

which are all defined in terms of the primitive constraints **union** and **complement**.

Figure 5.4 illustrates the network and the propagation of values. The network starts with the definitions for A_1 and A_2 as given, and proceeds to fill in as much of the table as possible. Propagation through the **union** constraint results in X_4 and X_5 being placed *out* of Y_1 and Y_2, by condition 3 of figure 5.3. Similarly, X_4 and X_5 must be *in* X since they are *in* A_2 and $X - A_1 = A_2$. These are the only conclusions possible from local propagation. Employing hypothetical reasoning, however, we are able to show by contradiction that X_1, X_2, and X_3 must be *in* X.

This leaves only six unknowns in the table, and therefore 2^6 possible bindings for the three theorem variables. This is a substantial improvement over the 2^{15} enumerations required by the most naive approach.[3] Of these 2^6 possible combinations, it turns out that only six are consistent with the network. Note that we do not have to enumerate even these 2^6, since the space is pruned as we notice contradictions

[3]The improvement is actually better, since the universe of attributes may have been larger than the five appearing in this goal. Such additional attributes would not contribute to the complexity of the constraint computation, but would enormously increase the cost of a straight enumeration.



Constraint Network	X_1?	X_2?	X_3?	X_4?	X_5?
A_1	IN	IN	IN	OUT	OUT
A_2	OUT	OUT	OUT	IN	IN
Y_1	?	?	?	out	out
Y_2	?	?	?	out	out
X	(in)	(in)	(in)	in	in

IN: given; in: derived from propagation; (in): derived by contradiction

Figure 5.4: Restricting possible bindings using constraint propagation

arising from partial assignments. As mentioned above, a mechanism that recorded dependencies would perform even better.

The interpretation is completed as we instantiate the theorem premise for each possible binding. At this point we notice that the six consistent bindings yield only three distinct axiom conjunctions. The resulting implications are added to RUP's database, and the appropriate subgoals are recorded in URP's control structure. The binding patterns are memoized, so that interpretation of the theorem for a goal with the same argument structure (for this example a **utility-independence** axiom with three attributes in the first argument and two in the second) may proceed without requiring any of this set-membership reasoning.

5.4 Control of Reasoning

Given a set of independence axioms and a goal axiom to establish, the control task faced by the system is to choose a path through the URP theorems that can prove the desired result. URP has no set overall control structure, since it has no set mode of usage. The default control structure can perform the sort of utility function choice task demonstrated in section 2.4. The knowledge structures and subgoaling (that is, theorem interpretation) procedures have been designed to work within a wide variety of conceivable reasoning strategies.

5.4.1 Default Control Mechanisms

Typically, the top level goal in reasoning with multiattribute utility theorems would be to find a functional form for a set of attributes (the *find* task), or to test the validity of a particular functional form (the *verify* task). URP maintains a goal structure, which is a tree when traversed in terms of subgoal expressions, but will generally contain cycles when traversed through individual axioms in the expressions. Goals may be proposed by the user, generated as subgoals during theorem interpretation, or suggested by other modules of URP itself. In a complete system, the mathematical reasoning facility would often request that a particular functional form be pursued.

Search of the goal graph may also be controlled in various ways. URP may proceed on its own based on a rudimentary best-first heuristic, by explicit direction from the user, or through other sorts of heuristic control. The implemented URP relies mainly on the best-first method, though it is relatively simple to exert manual control or to implement top-level driving procedures to focus the search according to some pattern.

Control for the *find* task is a bit more open-ended than for *verify*. Usually, a *find* task will contain several instances of *verify* tasks as subgoals. Under the default control procedure, URP first tries to establish that one of the n-attribute functional forms is valid over the whole universe of attributes. To do so, it successively applies *verify* to the axioms representing the multiattribute forms. In restricted situations, it will notice that certain forms are hopeless due to the absence of particular classes of premises. Increasing the sophistication of this mechanism would go a long way toward improving the efficiency of URP in the decomposition task.

5.4.2 Hierarchical Decomposition

Failing to find a direct decomposition after some effort (or failing to find a sufficiently simple one), the system uses forward reasoning to assert a two-attribute form from one of the independence conditions that is already known to the system (as a premise or by inference).[4] This decomposition form holds over two vector attributes corresponding to the arguments of the independence condition used to justify the form. For example, to use the independence condition

$$\text{(value-independence } (X_1\ X_2\ X_3)\ (X_4\ X_5))$$

for a binary decomposition, URP would introduce the two vector attributes $W_0 = X_1 \times X_2 \times X_3$ and $W_1 = X_4 \times X_5$. URP then pursues as subgoals the tasks of finding multiattribute forms for W_0 and W_1. If this search is unsuccessful, URP can try a different binary decomposition or pursue the n-attribute form further.

In restarting the search for a multiattribute form for the original set of attributes, it may be useful to automatically aggregate sets of attributes for which forms are known. For example, if a form for W_0 above was determined, our new problem is to find a form for $\{W_0, X_4, X_5\}$. Note that W_0 is treated just like its expansion in determining the truth value of axioms it appears in. However, the vector is treated like a single attribute for the purposes of set operations in theorem interpretation. The distinction between vectors and sets of attributes is defined more precisely in section 6.3.

[4]As described in the next chapter, several two-argument independence axioms directly validate two-attribute utility functions. These forms are given in appendix section E.1..

5.4.3 Heuristic Control

The possibility of providing some kind of declarative heuristic control language for defining search strategies has not been adequately explored. Such a capability would be desired, since the implemented scheme follows many obviously useless paths. There is reason to think that heuristics could help, because there is usually some structure to the kinds of axioms that are asserted, and to combinations of theorems that bridge commonly-occuring inference gaps. And providing this language should not be very difficult, since the theorem interpreter is coded to allow a fine-grained control over inference primitives.

One weakness of the implemented scheme is that the system cannot recognize when it is performing hopeless search. This is not a problem if axioms are actually asserted to be not true; URP will not bother trying to verify a proposition it can prove to be false.[5] The major difficulty is when there is a lack of asserted axioms. This is where forward-directed control and higher-level heuristic knowledge will become useful for containing the excessive backtracking, or at least for initially focusing the search.

Despite these problems, performance appears to be acceptable. In the examples tested, the system verifies true propositions with a reasonable amount of effort. Once a significant piece of the proposition space has been explored, retraction and addition of new axioms are handled quickly.

[5]It is not clear, though, that URP is as proficient at finding falsity proofs as at finding validating proofs.

6. The Multiattribute Decomposition Knowledge Base

One of the motivations for this project was the opportunity to develop a formalization of utility-theoretic knowledge in a uniform representation. In constructing the URP knowledge base, I have tried to capture this knowledge in a form that would lead to competent and efficient model construction while at the same time preserving the structure of the field as reflected in its literature. This chapter is an attempt to describe the contents of the portion of URP's knowledge base dealing with multiattribute decomposition. It serves to delimit the extent of the program's knowledge, with explanations for the terminology used and rationale for some important design decisions.

Because the discussion will be focused on issues of utility theory rather than on reasoning mechanisms or representations, the reader who is not primarily concerned with utility should not hesitate to skip this chapter. The intended audience consists of those who are interested in examining an encoding of utility-theoretic concepts, or others who may want to validate, correct, modify, or extend the URP knowledge base. It is provided to document the components of utility theory selected, and to justify the structures chosen to embody those components. While a knowledge-based system should ideally be "self-documenting," this is rarely the case in actuality, and URP is no exception.

A note on the status of the implementation: all of the concepts described in this chapter have been represented in the knowledge base (given in appendices B, C, and E), and a working interpreter for the theorem language exists. The capability of the system is described in sections 5.3 and 5.4. Achieving the performance specified in section 6.2.1 below would require further work.

6.1 Formalization

It may be argued that the knowledge of multiattribute utility theory is already formalized. The basic results of the field are expressed in mathematical terminology that is standard and unambiguous. Nevertheless, there is a great difference between formal description for human interpretation and formal description for automatic interpretation by computer. In writing a formal description for a human, we can be comfortable in assuming a certain forgiveness for minor abuses of the strictest conventions. Some of my points in this section and further below will have the flavor of nitpicking; in such cases recall that computers are the most notorious of nitpickers.

A formal description for automatic interpretation by computer must be coupled with a specification for a procedure which operates on that description. Any meaning assigned to the declarative representation only arises from how it is manipulated by that procedure. For our purposes, it will suffice to talk about the meaning of a representation in terms of the inferences that can be made by URP as a result of its inclusion in the knowledge base. To be clear about what is meant by an inference, I will try to specify precisely the classes of questions URP is expected to answer about each segment of its knowledge base.

This formalization exercise is a necessary part of building a system like URP. But the development of a utility-theoretic knowledge base has benefits independent of its use by a consultation program. At the very least it enforces consistent terminology for its concepts. Although this is usually a minor problem,[1] there is potential for confusion in concepts with more than one name (for example, value and additive independence, multilinear and quasi-additive forms) names with more than one concept, or similar/dissimilar names for dissimilar/similar concepts.

A related issue is the burial of knowledge in terminology. In a literature designed for human interpretation, part of the meaning of concepts may be contained in the names used to describe them. In a computer program, the name is just a symbol like any other, with no linguistic connection to other knowledge. All concept connections are explicit.

Another strict requirement for a computer representation is a consistent notion of data types. In the literature it is possible to refer to objects such as attribute sets or independence conditions without being perfectly precise about their semantic types. In the discussion below I will demonstrate the possible ambiguities or inconsistencies that may arise in such cases. Note that in general these ambiguities would present no problem to any reasoner with common sense (which includes most humans but not URP).

In developing a computer representation there is a strong incentive to keep the interpreter as simple and as general as possible. This in turn motivates the drive

[1] An inevitable one for any young, dynamic discipline. Utility theory probably suffers less from this malady than most other technical fields.

for a well-structured and uniform knowledge representation, since structure in the representation can often substitute directly for special-case mechanisms in the interpreter. This is desirable, because the declarative representation will be much more transparent than the interpreting procedure. Therefore, the representational formalism developed may contain structure which was not apparent in the original descriptions. An example of this is the means by which vector attributes combine with conditional independence concepts to yield hierarchical decompositions without special procedures for that concept.

The segment of utility-theoretic knowledge dealing with decomposition of multiattribute functions is the segment which I have had the most success in formalizing. This is undoubtedly due to its rich axiomatic structure—the characteristic which first suggested this undertaking, in fact. Experience in building this knowledge base should be useful for similar endeavors in other mathematical modeling disciplines.

6.2 The Extent of URP 's Knowledge

6.2.1 The Decomposition Task

The task performed by URP is best defined by a description of the class of questions it should be able to answer about multiattribute decomposition. The questions are expressed in terms of URP concepts (that is, axioms and attributes) and are answered with respect to the theorems contained in the knowledge base.

Questions are of the following general form:

Given:	A set of axiom assertions, G
Hypothesize:	A set of axiom assertions, H
Question:	Does G entail H? (written $G \vdash H$)

In the most usual case, H will represent some decomposition of the multiattribute problem culminating in a functional form or a hierarchy of forms. An axiom assertion is defined as a pair consisting of an instance of an URP axiom (where the arguments are of correct type) and a value of **true** or **false**. The sets G and H may be of any size. A question with a null H may be construed as "Is G consistent?" The analogous interpretation for a null G would be "Is H tautological?" This, however, would never be the case for a conjunction of URP axioms.

The main variation of this question form is to allow one of G or H to be unspecified or only partially specified. For unspecified H the task is to find a decomposition which is entailed by the assertions in G. The user may fix some constraints on the allowable decompositions, or specify optimality conditions. The most obvious optimality condition is to minimize the number of conditional utility functions and scaling constants to be assessed. Examples of prespecified constraints include:

- Restriction to single-attribute conditionals at base-level
- Prestructuring of hierarchy (specification of all allowable vector attributes)
- Maximum number of scaling constants and/or conditional functions

If the user leaves G unspecified, URP's task is to find a minimal set of axioms that would entail H. As above, the notion of minimality may be subject to some specification by the user.

A final variation would be to give URP two sets such that $G \vdash H$ and ask it to try to weaken G or strengthen H while still preserving the "$\vdash$" relation. This should be substantially easier than finding the minimal/optimal set in the first place.

I do not claim that URP will always be able to find these minimal Gs or optimal Hs, or even that it will even find any that exist. Though an exhaustive algorithm would always succeed in finding these, the space is so large as to require heuristic search techniques. An URP augmented to handle the "find" task should be good enough to answer questions of the first type with reasonable efficiency in the majority of cases, and should be fairly proficient at weakening/strengthening questions of the last type.

The final component of the decomposition task is explanation. Given a conclusion that $G \vdash H$, URP is able to trace its deductions, reporting the theorems employed at each step.

6.2.2 Coverage

URP's knowledge about multiattribute decompositions has been gathered from a variety of sources. The most important of these have been Keeney and Raiffa [61], Farquhar [27], Bell [6], and Fishburn and Keeney [36]. Other sources will be listed in the sections discussing their specific contributions.

It is difficult to characterize the subset of knowledge that is represented in URP. I claim that the program completely covers the decomposition results appearing in chapter six of Keeney and Raiffa's text, in the sense that it should be able to answer any "$G \vdash H$?" question that is answerable solely by applying theorems from that chapter. It should also be able to find answers to unspecified Gs and Hs at least as good as those findable from the text. All of the independence concepts mentioned there are expressible in URP, and all functional forms are represented. Hierarchical decomposition using conditional independence is performed without any prestructuring. Note, however, that I am not making any claims at the moment about URP's efficiency in performing this reasoning. And note that URP only performs the decomposition; it does not assess the scaling constants and conditional utilities necessary to completely specify the function.

The knowledge base also contains many concepts not appearing in Keeney and Raiffa. In principle, I believe that this knowledge covers any multiattribute decomposition that has been used in real applications to date (this does not include special-purpose decompositions for things like time streams). Currently, however, I do not claim completeness over any other part of the literature.

The knowledge base contains considerable redundancy. In general there are several paths to the same result (H), using different theorems or the same theorem instantiated for different patterns. Some theorems, in fact, could be removed without changing the class of decompositions that URP would be able to determine. These redundancies are retained because they often provide shortcuts in URP's inference, or refer to aggregate concepts that are useful in structuring preferences. I have generally taken the attitude that high-level theorems appearing in the literature deserve inclusion even if their content is subsumed by lower-level theorems because, after all, some utility theorist had a reason for thinking they were useful.

6.3 Attributes

In the standard notation and terminology, we say that the outcome space X is equal to the cross product of the sets of possible values for each attribute. That is, $X = X_1 \times \cdots \times X_n$. In this context it is clear that each X_i is a set corresponding to an attribute i. It would seem, then, that an attribute should be referred to by an index according to which dimension of the outcome space it corresponds. But in literature usage the symbol X_i is often used to refer to the attribute itself.[2] Here X_i is a name for the attribute, not a set. Using sets for names would present problems if more than one attribute had the same domain of values.

A collection of attributes may be referred to as a single *vector* attribute. The set of values possible for a vector attribute Y is the cross product of its constituent

[2] Or even to the *objective* for which the attribute is a measure.

attributes, $Y = X_{i_1} \times \cdots \times X_{i_k}$. But once again, we will take the symbol Y to be the *name* of the vector attribute only. For multiattribute decomposition, URP treats vector attributes exactly like scalar attributes.

It is important in URP to make a distinction between a vector attribute and a set of attributes. The problem arises in using theorems that perform set operations on the "arguments" of independence axioms. While conceptually the two objects are identical, in the course of its reasoning URP must distinguish between situations when a symbol Y stands for a vector attribute and when it stands for a set of attributes. Only in the latter case is it permissible to perform set operations on the object. Also note that while vector attributes may be nested (that is, it is permissible to have a vector containing vectors) arbitrarily, the members of attribute sets are all attributes, not attribute sets (though these attributes may be vectors.)

One limitation of URP that should be remedied is that it does not provide for *subrange* attributes. A subrange attribute can be used to indicate that a certain axiom is only applicable over a limited portion of the outcome space. Such assertions are a natural generalization of independence relations over entire attribute value ranges, and form the basis for the *multivalent* decompositions proposed by Farquhar [29]. Mechanisms for handling assertions of independence conditions over intervals would be similar to URP's facilities for handling assertions of qualitative behaviors over arbitrary intervals.

6.4 Independence Axioms

Probably the central knowledge base design decision in URP was the separation of knowledge about multiattribute decompositions into independence conditions, functional forms, and theorems relating them. The rationale for this structure is fairly obvious, given the large technical vocabulary developed for describing these conditions and the large number of papers in the literature which prove various theorems relating the conditions to each other and to functional forms.

Sorting through the literature, I found that a wide variety of independence concepts had been identified, that a few of these seemed most important (or at least most useful), and that there was some variation in the terminology or mathematical formalisms used to describe these concepts. This variation underscores the potential benefit to be gained from an effort to build a uniform representation.

An independence axiom is a specification for a relation defined over a fixed number of *arguments* which may be attributes or sets of attributes. The type and number of arguments varies among axioms. Note that in the literature these axioms are sometimes considered relations on subspaces of the outcome space, which corresponds to the view of attributes as value domains for dimensions of that space.

6.4.1 special-case-of

Many of the URP independence axioms are linked together by the **special-case-of** property. An axiom denoting relation A is a special case of the axiom denoting relation B if $A \subseteq B$. Note that this requires that the two axioms have the same number of arguments, and that the type restrictions on A are at least as strong as those on B.[3]

The **special-case-of** property is useful because it lets us make inferences from the hierarchical structure of the axioms without applying theorems. In defining theorems, therefore, we use the weakest version of the axiom in the premise and the strongest in the consequent. The results are always applicable for axioms stronger than the premise and/or weaker than the consequent.

6.4.2 rel-prop

The **rel-prop** list of an axiom contains the special properties of the relation defined by that axiom. Possible special properties include symmetry, transitivity, or other mathematical concepts of relations. These properties are not used very much in the URP knowledge base.

In sections below I will describe the particular axioms defined in URP. A complete listing of URP's axiom definitions is given in appendix B.

6.5 Theorems

An URP theorem specifies a pattern of implications between sets of axioms. Theorems are used to capture knowledge that links combinations of independence conditions to each other and to functional forms. The two main components of theorems, the *premise* and *consequent*, each represent a conjunction of axioms when the variables are bound to specific legal values.

[3]More generally, we would like to allow the arguments of B to be computable from A's, rather than requiring them to be identical. This would be useful in the case of conditional independence, for example, since there the axiom is related to a *class* of general cases.

Theorems in the URP knowledge base have been drawn from numerous sources. Many of these will be discussed in sections below, usually along with the axioms they deal with. A complete listing of URP's theorem definitions appears in appendix C.

Restrictions

Some theorems (and some axioms) are annotated with restrictions on the applicability of the result. These restrictions are informal and are never checked by URP. Usually they refer to highly technical conditions for which the knowledge base does not contain concepts. For example, many theorems and axioms require *essentiality* of the attributes (informally, that they matter to the decision), convexity of the space of attribute levels, or some other technical condition. I have not been consistent in recording all of these qualifications.

6.6 Multiattribute Functional Forms

Functional forms are symbolic specifications for the mathematical functions which represent different multiattribute decompositions. Each form is associated with a particular independence axiom whose truth value determines its validity for the arguments of that axiom. The forms can be instantiated for any number of attributes, including scalars and vectors. A complete listing of the functional forms in the URP knowledge base is given in appendix E. Recall that URP does not perform any reasoning based on the mathematical structure specified with those forms.

6.7 Preferential and Utility Independence

Preferential independence (PI) and *utility independence* (UI) in URP have the standard definitions given in Fishburn and Keeney [36], Farquhar [27], and elsewhere. Intuitively, (PI Y Z) holds if preferences for certain outcomes described by attribute Y do not depend on the fixed value of attribute Z. UI expresses the analogous condition for preferences over lotteries on Y. A more formal expression of these statements is given below in equations 6.1 and 6.2 on page 66. Members of the relation are pairs of mutually exclusive sets of attributes, where the first is PI (or UI) of the other. Utility independence is a special case of preferential independence.

Utility independence is directly connected to a two-attribute utility function, as described in Fishburn [34] and elsewhere.[4] Since sets of attributes may be translated to single vector attributes, each UI assertion can be a step in a hierarchical decomposition of functional forms.

PI and UI are the axioms appearing most commonly in URP theorems. Theorems 1, 2, 5, and 12 relate UI directly to aggregate conditions (see section 6.13 below). Theorems 9 and 33 build UI conditions from combinations of other UI conditions. Theorem 10 is the PI analog of theorem 9. Combinations of UI with weaker conditions are used to imply other UI conditions in theorems 18, 19, and 20. Weaker conditions (including PI) alone are sufficient for a UI conclusion using theorem 21. UI appears with implicit (strong) conditional UI (see section 6.10) in theorems 23, 24, 29, and 30. Theorems 25 and 26 define various relations among UI, PI, and their explicit (weak) conditional versions. The implicit and explicit forms of conditionalizing are related in theorems 27 and 28. Recall that these theorems can be found in appendix C in numerical order.

I have not included the concept of risk independence [58] in the knowledge base since it is a special case of utility independence for increasing, twice-differentiable scalar attributes that does not seem to offer any stronger conclusions or explanatory power.

6.8 Value Independence

Like PI and UI, *value independence* (VI) figures prominently in URP's knowledge base. The definition of VI is taken from Fishburn and Keeney [36], and is equivalent to definitions appearing elsewhere. Informally, VI holds between two attributes if only their marginal probability distributions are relevant. The concept of VI appears to be equivalent to additive independence as described in Keeney and Raiffa [61]. Value independence is a symmetric[5] relation on pairs of mutually exclusive attribute sets, and is a special case of UI (and of PI, etc., since **special-case-of** is a transitive property). This differs from usages of VI where the relation is over an unspecified number of attribute sets (such as the definition in Farquhar [27] and the usage of

[4] Bell [6] provides a very useful table which gives the most important two-attribute functions in conditional form, along with their associated independence concepts.

[5] In fact, VI is the only relation with a special **rel-prop** defined in URP. Relation properties, like **special-case-of** properties, simplify the reasoning by bypassing the theorem application mechanism.

additive independence in Keeney and Raiffa [61]). URP uses the name *additive independence* for the concept of VI over sets of (possibly vector) attributes.

Theorem 17 provides a way to build VI assertions from other VI assertions combined with a weaker axiom. VI is related to a functional form in theorem 22.

6.9 Joint Independence

In multiattribute decomposition, there is often special relevance to an independence condition that holds in both directions. Rather than define a special axiom and provide theorems defining such conditions, I have built a construct into the theorem language to facilitate expression of this concept. Wherever the qualifier "joint" precedes the name of an independence axiom, the expression denotes the conjunction of two relations—one for each ordering of the arguments. For example, the expression (**joint utility-independence** Y Z) can be considered an abbreviation for the two expressions (**utility-independence** Y Z) and (**utility-independence** Z Y).

6.10 Conditional Independence

Several independence axioms can be associated with conditioning concepts which enable consideration of independence conditions among non-exhaustive subsets of the outcome space. URP has axioms for conditional VI, UI, and PI.

Keeney and Raiffa [61] discuss two kinds of conditional independence. Consider a partition of the attributes into three nonempty sets, Y_1, Y_2, and Y_3. Saying that conditional "foo" independence (FI), for example, holds for Y_1 and Y_2 given Y_3 means that for *any* fixed value $y_3' \in Y_3$, Y_1 is FI of Y_2. An assertion that conditional FI holds for Y_1 and Y_2 given y_3' means that for the *particular* value $y_3' \in Y_3$, Y_1 is FI of Y_2. Clearly the first kind of conditional independence is a stronger assertion.[6]

In the URP knowledge base, conditional independence is associated with the weaker version. It appears to me that there is no real reason to treat the strong version of conditional FI any differently from FI in the context of a restricted universe of attributes. That is, any conclusions that are derivable from (**FI** Y_1 Y_2) in a problem where all the attributes are in Y_1 and Y_2 are also valid for a problem with additional attributes as long as it is understood that the result is conditional on

[6]Bell [3] means this strong version when he refers to conditional utility independence.

"the rest of the attributes." In other words, URP treats the implied conditional as if it were working on a problem of smaller dimension.

Conditional independence in URP, then, is a relation on three exclusive sets of attributes. The assertion (**conditional-FI** Y_1 Y_2 Y_3) means that *there exists some* $y_3' \in Y_3$ such that Y_1 and Y_2 are FI when the rest of the attributes are fixed at that level. For applying the theorems, it is not necessary to know what that level is, just that it exists. To assess the functions and scaling constants, however, it is necessary to know the conditioning value.

This treatment of conditional independence is not a completely satisfactory one because it does not truly represent the concept of conditionalization directly. It was necessary to include separate conditioning axioms for each independence concept, even though the notion of conditioning is orthogonal to the concept being conditioned. Nevertheless, this option was chosen over the more fundamental change in URP's framework that would be necessary to avoid the duplication. Reimplementations of URP should provide a better long run solution.

6.11 Weaker Axioms

In addition to UI, PI, and VI, Fishburn and Keeney [36] discuss four axioms that are strictly weaker conditions on preference interaction. Two of these, called *generalized* PI and UI (GPI and GUI), are slight modifications of the base axioms that allow for restricted kinds of choice reversals. Another paper by Fishburn and Keeney [37] discusses GUI in greater detail.

GUI appears as one of the consequents of theorem 19. It is used as a premise in combination with some PI conditions to conclude some UI conditions in theorem 21. Theorem 31 builds GUI conditions from other GUI conditions. GUI conditions alone are sufficient to validate a multiattribute form in applications of theorem 32. GPI axioms do not appear in any URP theorem.

The other two axioms, *indifference independence* and *weak indifference independence* (WII), concern riskless preferences and are less restrictive than any of the above. Indifference independence does not appear to be very useful (and does not appear in any URP theorems), but WII seems to be a valid substitute for PI in many situations.

WII is applied in theorems 17, 18, and 19. In each case, WII combines with stronger axioms (VI and UI) to conclude different instances of these stronger axioms.

6.12 Interpolation Independence

Interpolation independence (II), as described by Bell [4] [5] [6], is a relatively weak restriction on the interaction of preferences under risk that nevertheless provides substantial power in the decomposition of multiattribute utility functions. Briefly, II holds between Y and Z when the collection of conditional utility functions for y given z is related by a function of z alone. This condition is stated more clearly in the assertion of equation 6.3 on page 66. Joint II is directly related to a two-attribute decomposition which is defined in URP, and combinations of joint II conditions are sufficient to justify the multilinear generalization functional form (theorem 16).

Though URP is able to generate these II decompositions, it does not have the ability to execute the interpolation-based assessment procedure that Bell [5] advocates. This procedure provides a way to converge on increasingly accurate representations of the utility function for cases where II is only an approximation. URP does not provide for such approximation techniques, but it is easy to envision an adaptation which would accommodate their implementation. This point is discussed further in section 6.15.3.

6.13 Aggregate Conditions

All of the axioms discussed so far have been binary or ternary relations on sets of attributes. It is often useful to specify relations that hold over an unspecified number of arguments; for this purpose URP contains several axioms whose argument is a single set of attributes. As always, hierarchy is maintained by aggregating scalar attributes into vectors.

Many of these axioms relate directly to multiattribute functional forms. Theorem 1 describes sufficient conditions for applicability of the *multilinear* form. Sufficient conditions for the *multiplicative* form—a special case of the multilinear form—are given in theorems 3 and 32. The most usual justification for this form is *mutual utility independence* (MUI), which is a consequent of theorems 2 and 5. MUI can be used to conclude individual UI conditions by application of theorem 12.

The additive independence axiom does not appear in any theorems, but is the precondition for applicability of the additive form for the multiattribute utility function. Additive independence is a special case of MUI.

Theorem 14 relates *generalized bilateral independence* to the *bilateral* form, but these concepts are not well-developed in URP. The same goes for the concept of an *additive value function*, the consequent of theorem 13. Finally, the *multilinear generalization* form is valid under combinations of joint interpolation independence conditions, as defined in theorem 16.

6.14 Axioms not Implemented

There remain a few URP axioms that do not figure in any of URP's reasoning, but are retained for future expansion of the knowledge base. It is expected that indifference independence, GPI, *bilateral independence*, *parametric independence* [63], and *mutual preferential independence* will become useful as the URP knowledge base is developed.

A further gap exists in independence conditions and decomposition approaches that are entirely absent from URP. Of these, the multivalent structures [29] mentioned above and fractional hypercube models [26] probably merit some attention. Another possibility is risk invariance as developed by Willig [113]. The additive and multiplicative generalizations described by Bell [5] would also be useful, but URP cannot express the preconditions concerning corner constants.

The multiattribute decomposition knowledge base (as well as the rest of URP) is in a prototype stage. I like to view the implemented version as only a basis for a potentially more solid and complete encapsulation of important results from utility theory.

6.15 Assessing the Decomposed Function

As mentioned earlier, URP contains a substantial amount of expertise in choosing a multiattribute decomposition, but almost none regarding assessment of the resulting function. In order for the program to be useful to a utility analyst, it should provide a lot more help in this portion of the model construction task. The absence of this capability in the implemented URP system is due to lack of time; the present framework should be able to accommodate the necessary knowledge. In this section I examine the possibilities for inclusion of better multiattribute assessment facilities within URP. Issues of single-attribute assessment are the subject of chapter 7.

6.15.1 Assessing Scaling Constants and Conditional Functions

The only representation of scaling constants and conditional functions in URP is in the specification language for multiattribute functional forms. The specification only defines these objects as components of the multiattribute decomposition, specifying their values in terms of lower level objects such as evaluations of the function at specific points. The specification does not provide any clue about how to go about finding or constraining the values for these objects.

An automatic assessment facility would model-independent definitions for these components, including partial procedures for determining their values. Assessment of conditional utility functions should be a straightforward modification of the methods for single attribute assessment described in chapter 7.

6.15.2 Relation of Axioms to Conditional Functions

As far as URP is concerned, independence axioms are only important to the extent that they may be applied in decomposition theorems. Since these axioms are often defined in terms of what they imply for conditional preference orderings, it seems that we are missing important information by limiting their use in this fashion. For example, the proposition (**UI** Y Z) has the assertional import of

$$\forall y \in Y, \forall z', z'' \in Z\,[u(y|z') = u(y|z'')] \tag{6.1}$$

An assertion of (**PI** Y Z) means

$$\forall y', y'' \in Y, \forall z', z'' \in Z\,[u(y', z') > u(y'', z') \Rightarrow u(y', z'') > u(y'', z'')] \tag{6.2}$$

Finally, (**II** Y Z) is true if and only if

$$\forall y \in Y, \forall z, z', z'' \in Z\,[u(y|z) = \theta(z)u(y|z') + (1 - \theta(z))u(y|z'')] \tag{6.3}$$

from Bell's definition [6]. Other independence conditions have implications for conditional functions of a similar structure. These conditions should lend substantial constraint to the values permissible for these functions.

6.15.3 Encoding of Assessment Algorithms

In section 6.12 I pointed out that the URP knowledge base does not capture assessment procedures that are often associated with particular decompositions.

This has been partly intentional, since an aim of this project has been to avoid rigid algorithms in favor of more flexible use of preference specification. Nevertheless, flexible use of information does not imply that there should not be more focused information-gathering techniques.

There may be natural ways to implement multiattribute assessment algorithms within URP's framework through the introduction of question-asking heuristics. Sets of heuristics associated with particular model situations may be switched in and out to provide for the "execution" of a procedure. Any such assessment framework must be integrated with the single-attribute assessment concepts developed in chapter 7, below.

7. Assessment with Preference Choices

7.1 Model-Independent Interpretation

Traditional utility assessment relies heavily on expressed preference choices to determine the parameters of a utility function. The mechanisms built into URP deal exclusively with reasoning about the form of the utility function, which is usually a precompiled component of assessment tools. URP's flexibility in model structure affords a less-constrained approach towards assessment, where hypothetical preference choices are interpreted under a procedure which is independent of a particular underlying utility model. Assessment programs that employ a prechosen model generally implement a procedure for generating lottery questions that fill in the parameters of the model with the fewest possible responses. The interpretation of the response is a direct translation to a particular parameter of the model.

In URP, interpretation of preference choices must be more general. Since the model structure may be changed at any time (due to assertion or retraction of independence conditions or qualitative behaviors), URP must be able to compute the implications of a hypothetical preference choice for any utility model. The extra generality means that URP will not be as good at optimizing the set of questions generated for particular models, since programs using static models can employ efficient question-asking strategies that exploit the special structure of their particular models. This may not be as important as it sounds, however, because URP does not require completely specified utility functions. The completeness requirement of traditional assessment tools places an unfortunate emphasis on the coverage of a set of questions, at the expense of their information content with respect to a particular decision, or consideration of how easy they may be to answer. In general, these efficient question sets include lotteries involving outcomes at the extreme ranges of the attributes, which are considered difficult to answer and prone to biased responses.

In the remainder of this chapter I outline a framework for utility assessment using hypothetical preference choices in URP. Keep in mind that this chapter does not describe an implementation, rather it conveys an overall attitude and some intriguing assessment capabilities which may be possible under the URP approach.

7.2 Inequalities vs. Indifference Points

A *lottery* is a choice presented to an assessor between two prospects, each a probability distribution over outcomes. I will use the notation $L_1 \succ_c L_2$, where the "$\succ_c$" relation is read "is chosen over,"[1] to describe the assertion that the assessor expressed a preference for lottery L_1. The lotteries offered to the assessor are almost always simple—one or two possible outcomes in each L—to limit the processing burden on both the assessor and the program. Complicating the outcome distributions does not provide any information advantage to the reasoner.

A common device used by assessment procedures is the certainty or probability equivalent, where the subject is given two lotteries (L_1 and L_2) with a single parameter (a probability or outcome from one of the Ls) unspecified, and asked to indicate the value which would make her indifferent between L_1 and L_2 ($L_1 \sim_c L_2$). Note that an answer to this sort of question yields more constraint than a preference choice, since it specifies an equality point. But by dragging the assessor all the way to the indifference point (perhaps converging from either side), we are straining the limits of her judgmental confidence. We would expect these questions to be more difficult and prone to errors, as studies have shown [48]. Moreover, without the restriction to fully specified utility functions, indifference questions are often unnecessarily precise. Indeed, assessment algorithms generally stop after the minimal number of questions are asked, because further questions inevitably lead to contradictory conclusions. In such cases it is necessary to resort to statistical best-fit schemes to assign values to parameters based on lottery choices. Due to these concerns, assessment in URP is restricted to inequality questions, with the required narrowness of bounds determined by the particular decision at hand.

7.3 Constraints on Utility Functions

Information in the form of hypothetical preference choices constrains the space of possible utility functions by revealing portions of the underlying preference relation ($\succ$). By assuming that $\succ$ is the same as $\succ_c$ (an assumption of every utility assessment program), each preference choice yields a linear inequality among points

[1] The subscript c is used to distinguish this relation from the usual preference relation "$\succ$," meaning "is preferred to." I make this distinction to allow for the interpretation of elicited responses under a descriptive model other than expected utility, discussed in section 7.4 below.

on the utility function. That is, an assertion that $L_1 \succ L_2$ implies

$$\sum_{x \in X} p_{L_1}(x)u(x) > \sum_{x \in X} p_{L_2}(x)u(x) \tag{7.1}$$

where p_{L_1} and p_{L_2} are probability mass functions corresponding to the two lotteries. Naturally, $\sum_X p_L(x) = 1$ for all lotteries.

Without any further information about u, we can use these linear constraints to compute bounds for the utility of specific outcomes (those appearing in at least one lottery), or distributions over outcomes. Treating each evaluation of u at a point (generically denoted by "$u(\cdot)$") as a variable, we calculate the lower bound for an expression made up of a linear combination of utility values by solving a linear program. The objective function to minimize is simply this expression, subject to the constraints derived from the lottery choices and equation 7.1, augmented by constraints of the form $0 \leq u(\cdot) \leq 1$. To compute the upper bound, we minimize the negation of the expression.

Incorporating other kinds of constraints on u complicate the computations. Some of these, such as monotonicity of u with respect to scalar outcomes, can be represented as linear constraints (not necessarily very efficiently), while others require nonlinear representations. Combining preference choice information with specific functional forms for u will generally result in a set of nonlinear constraints. Computing bounds for nonlinear expressions subject to nonlinear constraints is a much harder problem, closely related to the dominance-proving problem discussed in section 4.1.2. Algorithms for restricted versions of the problem exist, but I have not explored the possibilities for incorporating them in URP.[2] Naturally, the most general methods are least efficient and least inclined to converge, while more powerful techniques only work on restricted classes of problems. Choosing a constraint reasoner for URP will involve classifying assessment subtasks according to the constraint types they involve. This is an important focus of the effort required for extending URP to perform utility assessment.

[2]One example is the algebraic constraint manipulation system developed by Brooks for geometric reasoning in vision [17], which is based on a linear procedure called SUP-INF originally due to Bledsoe [13] and improved by Shostak [96]. While the usefulness of this system for the kinds of problems that would come up in URP is untested, I suspect it will be able to reason about some of the multiattribute forms: bounding multiattribute utility from single-attribute lotteries and vice-versa. It should also be able to handle some of the nonlinearities arising in some of the descriptive theories described below. It appears, however, that this system would be incapable of reasoning about some of the more important single-dimension utility functions.

7.4 Alternate Descriptive Theories

The descriptive validity of expected utility theory has been a subject of much controversy ever since the development of the model over forty years ago. As Schoemaker points out in his review of the field [93], researchers with different perspectives on or purposes for the model have diverging views on the validity and importance of the descriptive aspects of expected utility. While it is legitimate to support the prescriptive significance of the model without making any descriptive claims, practical uses of the model for decision making invariably require subjective preference judgments from an individual to determine the utility function. By incorporating these assessed preferences in a formal utility model, the analyst is making an assumption about the relation between professed preferences and utility: in effect adopting a descriptive view of expected utility.

This prescriptive-descriptive cycle is disturbing, if not embarrassing. Analysts tend to support their formal models as normative approaches toward overcoming the failings of human judgment, yet their own models depend crucially on the ability of humans to make certain judgments. The only way out is to demonstrate that the required judgments are significantly simpler and less prone to error than the original decision problem. But simpler may not be simple enough; empirical studies using choices similar to those used in utility assessment consistently reveal important cognitive biases.[3] As Tversky [105] forcefully argues, decision analysts need to worry a lot more about this.

In the remainder of this chapter, I will discuss some of these biases and how the URP framework offers an interesting way to address the problem. The basic idea is to exploit URP's flexibility to interpret preference choices under descriptive theories other than maximization of expected utility. That is, we do not need to assume that $\succ$ is equivalent to $\succ_c$. Instead, we define some other relationship between $\succ_c$ and $\succ$, corresponding to a particular psychological theory of preference choice. Though the extra degrees of freedom often introduced by these alternate theories will generally weaken the conclusions we can make from a given set of hypothetical choices, I suspect that these should still provide useful constraint on the utility function.

[3] See Tversky and Kahneman [106] and Hershey, Kunreuther, and Schoemaker [47] for a general flavor of the results.

7.4.1 Descriptive Theories of Preference Choice

Psychologists, decision theorists, and economists have long noted discrepancies between subjects' expressed preference choices (in the laboratory or the real world) and normative decision theories. Over the years a substantial literature has developed, cataloging a large number of persistent biases observed in typical utility measurement tasks. While identification of these cognitive effects is useful for developing bias-resistant assessment procedures, a more significant benefit may be reaped through the development of formal descriptive theories of preference choice. The remainder of this section outlines some of the potentially useful theories developed over the last few years; my focus is on the structure of the theories rather than their rationale or descriptive validity. The description may get overly technical at times, since illustrating the feasibility of turning these diverse theories into assessment procedures requires a demonstration that their mathematical structure may be incorporated into the URP framework.[4] Implementation of interpreters for them is discussed in the next section (7.4.2).

Prospect Theory

Prospect theory [55] attempts to account for observed violations of expected utility in expressed preference choices with a two-step process of editing and evaluation of simple lotteries. The first phase is an editing procedure, where the lotteries are transformed in ways which may or may not preserve their normative utility-theoretic properties. Kahneman and Tversky describe six editing operations, each a straightforward syntactic manipulation of the lotteries. The first operation, *coding*, requires identification of a reference point, which dictates the perception of an outcome as a gain or loss.[5] Some of the other editing operations are valid manipulations under expected utility, such as the removal of common subcomponents. This is equivalent to the transformation described earlier by equation 4.2.

[4]I would suggest that the reader skim over the mathematical details that are not of major concern while paying special attention to the *form* of the models. The individual theories are much better motivated in the cited papers than in the presentation here.

[5]Fischhoff [33] notes that it may not always be possible in practice to determine the reference point used by assessors in this editing step. In particular, it seems that subjects' own impressions of their reference point is often misleading (at least under a prospect theory interpretation). This may be a hindrance to its use for assessment, since interpretation of preference choices under prospect theory will require us to identify the reference point.

Evaluation of the edited prospects requires a weighting function of probability, π, and a value function, v. The value of a prospect with two nonzero outcomes x and y, associated with probabilities p and q respectively is given by

$$V(x,p;\ y,q) = \begin{cases} \pi(p)v(x) + \pi(q)v(y) & \text{if } p+q<1 \text{ or } x \geq 0 \geq y \\ & \quad \text{or } x \leq 0 \leq y \\ v(y) + \pi(p)\left[v(x) - v(y)\right] & \text{if } p+q=1 \text{ and either} \\ & \quad x > y > 0 \text{ or } x < y < 0. \end{cases} \tag{7.2}$$

The theory also dictates certain constraints on the functions π and v. For example, subjects tend to exhibit risk aversion for gains and risk proneness for losses, and losses loom larger than gains. Thus, $v''(x) < 0$ for positive x, $v''(x) > 0$ for negative x, and $0 < v'(x) < v'(-x)$ for all x (if these derivatives exist).

As for the probability weighting function, prospect theory asserts that π is increasing in p, with $\pi(0) = 0$ and $\pi(1) = 1$ (though the endpoints may not be continuous). In addition, $\pi(p) > p$ and $\pi(rp) > r\pi(p)$ for small p and $0 < r < 1$. For all p, $\pi(p) + \pi(1-p) < 1$ (subcertainty).

Karmarkar's subjectively weighted utility model [56] evaluates lotteries by combining a classical expected utility function with a transformation of the probabilities. Unlike prospect theory's $\pi(p)$, Karmarkar's probability weights are computed using a function of a single parameter. This restricted form may make assessment considerably more tractable.

While discussion thus far has been confined to single-attribute lotteries (as are assessment tools and descriptive theories in general), some of these concepts may be extensible to multidimensional problems. For example, Payne, Laughhunn, and Crum [84] discuss ways in which editing operators similar to those of prospect theory may be applied to multiattribute lotteries.

Regret Theories

It has long been observed that decision makers may derive value or disutility from the correctness of their decision, separate from the utility attributed to the outcome itself. This phenomenon—called *regret* or *rejoice*—has been used to explain deviations from expected utility theory, and has recently been the centerpiece of some formal models of preference choice.

Bell's approach [8] is to model utility as a two-attribute function of final assets x, and foregone assets y. A series of assumptions about preferences for the two attributes leads to the result

$$u(x,y) = v(x) + f(v(x) - v(y)) \tag{7.3}$$

where v is a function measuring the incremental value of assets and f is an arbitrary function. Further qualitative characteristics of regret constrain the space of possible fs. Note that direct use of this two-attribute function requires that the joint probability distribution for x and y is specified (that is, outcomes of foregone lotteries are resolved). In a sequel paper, Bell [10] examines the special case of independence between lotteries, and cases where foregone lotteries are left unresolved. In using regret theory for assessment, it is necessary to choose a policy regarding the resolution of unchosen lotteries.

The regret theory of Loomes and Sugden [71], while not founded on a two-attribute utility-theoretic model, yields implications similar to Bell's.

Consumer Choice Models

Researchers in marketing are interested in a descriptive model of preference choice to predict consumer behavior. While much of their work involves stochastic choice, or choice over aggregate populations, some of their models may be applicable to deterministic individual behavior (in a modified form, if not directly). Of particular interest are those that have a utility-theoretic basis (for example, the approach described by Hauser [46]). Possibilities for assessment can be envisioned for those theories employing specific utility forms in conjunction with measurement error models (that of Eliashberg and Hauser [25] provides a recent example). While these models may not have a compelling psychological rationale, they may have value for assessment. Treating these models as descriptive theories and using them for assessment is similar in spirit to statistical best-fit approaches to utility modeling.

Accounting for Disappointment

The final example of a descriptive theory that may be used for assessment is Bell's recent model of disappointment effects [11]. Disappointment is conceptually similar to regret in that preference for outcomes is dependent on structural features of the lottery. The major difference is that here the relevant comparison is to expectations, not foregone assets. Decision makers experience disappointment when outcomes are significantly worse than expectations, and elation when outcomes are better. These measures of psychological satisfaction form a second component (in addition to the outcome itself) of utility for a lottery.

Consider a lottery that offers a probability of p of obtaining outcome x and $1-p$ for a less-preferred outcome y. In the simplest model—where preference for the outcome and for psychological satisfaction are additive, expectations are linear in the outcome, and disappointment-free preference is risk neutral—the certainty equivalent is

$$y + (x - y)\pi(p) \tag{7.4}$$

where π may be interpreted as a subjective probability measure.[6] Using this simple model, preference choices yield linear constraints on $\pi(\cdot)$, resulting in straightforward assessment. Bell [11] discusses direct assessment of π by indifference points, as well as less direct methods requiring explicit tradeoffs between the outcome and disappointment/elation. This latter form of assessment employs questions regarding the value of winning (or losing) lotteries instead of straight preference choices.

7.4.2 Interpreting Choices with Respect to a Theory

The key thing to notice from the above descriptions is that several of these alternate theories of preference choice have analytic forms that may be related to traditional prescriptive models. To implement an assessment procedure based on one of them rather than maximization of expected utility, we would simply replace the utility maximization criterion (equation 7.1) with the choice criterion operating in the descriptive theory (equation 7.2 for prospect theory, 7.3 for regret theory, or 7.4 for disappointment). An expressed choice of one lottery over another yields an inequality within that particular formalism; constraint reasoning analogous to that described in section 7.3 above may be applied to infer facts about the model's components.

Example: Implementing Assessment Based on Prospect Theory

An expressed choice in prospect theory constrains the possibilities for v and π, beyond the restrictions provided by the theory. URP's mechanisms for reasoning about partially specified utility functions may be applied directly for dealing with

[6]Note the similarity to prospect theory. Even though prospect theory's π arises from behavioral considerations of probability interpretation, when combined with editing of risk-free components it results in the same model as disappointment (see the second clause of equation 7.2), ignoring the transformation by v.

incomplete representations for functions like these. To derive a prescriptive model from our choice-constrained descriptive model, we need to apply transformations specific to the particular theory. For prospect theory (and similarly for Karmarkar's weighted utility [56]), we might replace π with the identity mapping, in effect treating v as the utility function. In addition, we probably would eliminate the prospect editing operations that are not utility-preserving. Deciding how (and whether) to adjust the reference point is a tricky question. This adjustment would have a serious impact on v, which should be appropriate in many situations.

To illustrate assessment under prospect theory, consider an expressed preference for the edited prospect $L_1 = (x_1, p_1;\ y_1, q_1)$ over $L_2 = (x_2, p_2;\ y_2, q_2)$, with $p_1 + q_1 < 1$, $p_2 + q_2 < 1$, x_1 and x_2 positive, and y_1 and y_2 negative. From this we may conclude (from 7.2)

$$\pi(p_1)v(x_1) + \pi(q_1)v(y_1) > \pi(p_2)v(x_2) + \pi(q_2)v(y_2) \tag{7.5}$$

which is a nonlinear constraint on eight values. To simplify this for assessment, we would typically ask questions in which some of the ps and qs or xs and ys are the same (or known points of the functions, such as $\pi(0)$, $\pi(1)$, or $v(0)$). For example, suppose $x_1 = 300$, $p_1 = .1$, $x_2 = 100$, $p_2 = .2$, and $y_1 = y_2 = 0$ in 7.5. This constraint, together with prospect theory's subproportionality and subadditivity conditions (which imply that $\pi(rp) > r\pi(p)$ for $0 < r < 1$) result in the relation $v(300) > 2v(100)$, the result that would be concluded from expected utility theory. In addition, the condition that v is concave for positive x enables us to strengthen our result.

$$\frac{1}{3} \leq \frac{v(100)}{v(300)} < \frac{1}{2}$$

Additional prospect choices add further constraint on the shapes of v and π. A constraint reasoner that is able to handle limited nonlinearity (such as the one Brooks developed for ACRONYM [17], noted above) will be needed to perform this task in any useful assessment tool.

Choosing a Descriptive Theory

Once several assessment procedures based on different descriptive theories have been implemented, we need to decide which interpretive model to use in which situations. Ideally, we would like to "diagnose" the types of bias that are present for particular assessment sessions and apply the theory that best explains that flavor of bias. This is certainly beyond the state-of-the-art, though I suspect that the capability of switching between alternate theories may provide a useful tool for bias diagnosis. But even without such sophistication, it may be possible to match descriptive theories to assessment situations based on other criteria. For example, the type of attribute being assessed (financial, length of life, health dimension) may indicate that certain psychological choice theories are more relevant than others. Most of the theories developed are specifically geared toward preference for financial assets, though other work has looked at related biases for other attributes. An example is the study of McNeil, Pauker, Sox, et al. [76], which found correspondences between biases due to framing of health outcomes and prospect theory.

Another possible criterion is the setting for collecting the preference choices. While the theories described above are generally related to answers to lottery questions similar to those used in traditional utility assessment, often it is desirable to use preferences revealed through real-world observation of decision making (such as consumer behavior, safety practices, public policy, or other economic activities). In such cases it would be a good idea to consider some of the biases typically observed in real-world economic behavior; the marketing models mentioned above may be particularly suited to these situations. Thaler [102] describes many biases that seem to be widespread in economic behavior. One of them, due to the phenomenon of *self-control*, is incorporated in a model developed by Thaler and Shefrin [103]. Weinstein and Quinn [109] discuss biases which are prevalent in societal health policy.

Conclusion

By representing choices in a theory-independent manner, we may switch between interpretive frameworks in a modular fashion. Thus, this psychological modeling serves as a form of structural sensitivity analysis, in which we weaken the assumption of normative choice to examine its effect on the result. If the decision proves invariant under a set of theories that admit various sorts of response bias, we may be justifiably confident that our result exhibits a novel form of robustness that would be difficult to demonstrate any other way.

8. Incorporating Domain Preference Knowledge

In a previous section (3.4) I put forth the contention that the qualitative properties used by URP are "one step away" from being intuitive. In this chapter I argue that the reason the concepts used by the program are not totally intuitive is that there is inevitably a substantial gap between general knowledge about a modeling discipline and knowledge of how to apply its models in a particular problem domain. This difficulty is quite serious, since a consultation system that knows everything about a field of mathematical modeling but nothing about relating those models to particular real problems would be virtually useless to practitioners trying to solve these real problems. Nevertheless, we do not wish to create narrow systems designed for modeling problems in only a restricted domain. The aim is to separate general modeling knowledge so that it can be employed in problems from a wide variety of areas.

The sections below serve to define the nature of the gap between modeling and domain knowledge more clearly. I provide a framework for spanning URP's gap by incorporating domain preference knowledge, illustrated by examples relevant to medical decision problems.

8.1 Modeling and the Ground Domain

The use of a mathematical modeling technique always involves an area of application separate from the modeling discipline. Gale and Pregibon [43] refer to this area as the *ground domain.* They point out that relating modeling concepts to experts in a ground domain is a serious challenge in building knowledge-based systems for statistics. The approach taken here is somewhat different: rather than building this relation on a user interface, the idea is to bring concepts from the domain into the modeling knowledge base.

8.1.1 Modeling as Abstraction

Whenever a mathematical formalism is used to model real world phenomena—

for decision making, prediction, theory formation, or whatever—there is inevitably an abstraction from the objects of the real-world domain to objects in the mathematical domain. The modeling exercise is appropriate if and only if the abstraction can be said to preserve the *essential* properties of the real-world object, where essentiality is defined relative to the goals of the modeling effort. Since this requirement is central to any modeling discipline, such disciplines have generally evolved rich vocabularies for specifying the essential properties and relating them to the modeling tools which comprise the discipline. The essential properties are commonly referred to as modeling assumptions.

8.1.2 Technical Vocabularies

For modeling disciplines intended for application in several domains (or even to varying problems in a single domain), the vocabulary must be in a language with a technical level more akin to the objects of the mathematical formalism than to the real-world objects being modeled. This is why a system like URP, which contains a substantial amount of knowledge about modeling and none about a ground domain, must deal in concepts that are one step away from those that would be intuitive to an application expert using the program. It seems ironic that the same generality of the essential properties that made them qualitative and therefore closer to intuitive principles in the first place also provides the argument that they cannot be intuitive in the final sense.

I have spent the greatest portion of this thesis describing what I think the *technical vocabulary* for preference modeling using utility theory looks like, and how it is possible to get a computer program to use it. But while utility theory provides an exemplary case of the vocabulary notion, the idea is by no means limited to this particular theory or this particular modeling task. I will further contend that *any* modeling formalism designed to work for more than a single problem instance needs a technical vocabulary.

Perhaps this point will be strengthened with a few brief examples. In probability theory (a component of many modeling disciplines), there is a fundamental abstraction from real-world phenomena to a formal notion of events in a space of events. Modelers use tools of probabilistic analysis to make conclusions about the real world phenomena represented by these abstract events. The technical vocabulary in this case includes (among other things) the characterizations of events which make various tools and methods appropriate or inappropriate. Some very simple characterizations (of collections of events) include mutual exclusion, collective exhaustion, and various flavors of statistical independence. Note that few programs that perform so-called "probabilistic reasoning" have explicit knowledge of even these simple characterizations.

The second example is in the area of statistical test design. Although this modeling task presents several important issues, one of the most fundamental ones concerns the measurement scales of the parameters of the model. In designing a test sequence, it is essential that the modeler (human or computer program) choose only tests that manipulate the parameter in ways meaningful with respect to measurement properties of its scale. To a statistician, the concepts of ordinal and interval scales, for example, are very basic and simple to apply. The scale type for a particular parameter can be easily identified, and tests appropriate for that type can be generated. Nonetheless, the vocabulary is technical, and there does not appear to be a simple one-line explanation that would enable an expert from any domain to determine whether her scale is ordinal. A human statistician usually can solve the problem by resorting to common sense or world knowledge; often it is not necessary to be an expert in the ground domain to elicit the essential properties of the modeled domain objects.

The two examples described were intentionally taken from extremely simple modeling issues (the second issue—reasoning about measurement properties—will be discussed in the context of utility modeling later in this chapter). The point is that if even the most simple of modeling concepts require a technical language, the more sophisticated modeling concepts will undoubtedly be beyond intuitive explanation for a particular ground domain.

8.1.3 Expressing Domain Concepts in a Technical Vocabulary

Even with this serious limitation, reasoning in terms of qualitative properties is substantially more explainable than strict quantitative reasoning. The vocabulary is still within the technical bounds of mathematics and the modeling discipline, but it is a lot closer to the fringe. Thus, the models will be accessible to those with less modeling expertise and perhaps more comprehensible even to modeling specialists. And for application to particular ground domains, or restricted problems within domains, it may be feasible to build in domain-specific knowledge about the mapping from domain principles to this technical vocabulary.

This last option is particularly reassuring for incorporation of preference modeling in decision-making knowledge-based systems. These programs typically operate over a narrow range of problem types in a highly restricted domain. I suspect that it will often be reasonable to build in the ground domain concepts necessary to maintain an interface to URP in terms of its own technical vocabulary.

8.2 A Domain Preference Knowledge Base

To test the expressive power of URP's technical vocabulary, I suggest the development of a domain preference knowledge base: a collection of useful knowledge about modeling preferences in a particular domain encoded for use by URP. This knowledge base would act as an interface module for programs using URP and as a library of models and components for a human analyst using the program.

In the remainder of this chapter I provide a framework for building this knowledge base using examples from preferences in medical decision problems. None of these representations and mechanisms have been implemented, but the design is compatible with URP's overall framework. I expect that many of the concepts proposed for inclusion in the knowledge base can be incorporated into URP with little modification of the existing system. Others will require more substantial implementation effort. The knowledge base for medical preferences as developed in this chapter is summarized in appendix F.

8.3 Built-in Attributes

8.3.1 Specifying Attributes in URP

The first step in encoding domain preference knowledge is to specify attributes that will be relevant to decisions in that area. Patterns for combining these attributes constrain the base structure of preference models, and URP assertions about their utility-theoretic properties help determine the form of these models.

Attribute Definition

An attribute definition in URP consists of a name, a measurement scale, and a description. Figure 8.1 depicts a sample URP attribute definition for the *life-years* attribute used in the medical example of chapter 2. The name "LIFE-YEARS" is simply the symbol used by URP to refer to the attribute, and **value-set** defines the set of values possible for that attribute. This set may be any collection of real numbers or integers (not necessarily contiguous).

```
(defattribute LIFE-YEARS
    :value-set '(real-range 0 max-life-years)
    :scale-type 'ratio
    :unit 'years
    :description
       '(life-years (# of years from NOW until DEATH)
         max-life-years (Largest possible value for life-years)))
```

Figure 8.1: An example of an attribute definition in URP

The **scale-type** specifies some measurement-theoretic properties of the attribute scale. Given a set of values and some interpretation of ordering and combination (generally some kind of addition or concatenation) associated with the attribute quantity, the scale-type defines the set of mathematical transformations that preserve this interpretation. Table 8.1 associates some of the most common scales with their class of permissible transforms.[1] This information is highly useful in model-building, since the class of transformations constrains the kinds of concepts that make sense for a particular attribute. For example, it is reasonable to talk about direction of preference along all of the depicted scales (except arbitrary naming), since direction is invariant under all monotonic transformations. It is dangerous to consider the convexity (risk properties) of preference for attributes with ordinal scales, however, since this property may be changed by nonlinear transformations. While consideration of risk aversion for attributes defined by log-interval scales is not a good idea, the risk property of the logarithm of that attribute may be a useful concept.

Measurement properties are likely to be useful in several components of the overall preference modeling task: formulation and explanation are two that come to mind. In general, I would argue that the measurement properties of the quantities manipulated by a mathematical modeling program are an essential part of the knowledge required to reason about them, and should therefore be included

[1] The volume by Krantz, Luce, Suppes, et al. [66] is the source for these definitions (except arbitrary naming), and is a good general reference on the theory of measurement.

scale-type	permissible transformations
arbitrary naming	all transformations (since no ordering exists)
ordinal	all monotonic transformations
interval	$\alpha x + \beta,\ \alpha > 0$
log-interval	$\alpha x^{\beta},\ \alpha > 0, \beta > 0$
ratio	$\alpha x,\ \alpha > 0$

Table 8.1: Some useful measurement properties and their permissible transformations (x denotes the measured quantity)

in any system that works with different kinds of quantities. I have already noted the usefulness of measurement properties to determine the applicability of statistical tests. Specification of the measurement scales provides a convenient way to indicate something about the allowable intuitive interpretations of a quantity's magnitude without a deeper knowledge of the real-world concept associated with the quantity. In other words, it describes the structure of the informal interpretation of the quantity without requiring that the computer understand what that intuitive meaning is.

The scale specification may need to be fairly complex, because many attributes are described by membership in arbitrarily defined categories rather than by continuously varying quantities, discrete gradations on a continuous scale, or the cardinality of a set. The **unit** slot of an URP attribute definition measured by a quantity is the semantic dimension of that quantity; for attributes measured by category membership, **unit** specifies the mapping from the attribute's value to definitions for the various categories represented. In the categorical case, we can still talk about the properties of numeric functions by associating an index set of integers with the set of categories. Recall that the main purpose of defining attributes is so that we can specify utility-theoretic properties of preferences over them (discussed further below). Therefore, the indexing scheme should be chosen with these properties in mind. An obvious desideratum in specification should be to choose a scale which provides the maximum structure for interpreting the value with respect to preference. Keeney [59] discusses issues in selecting and constructing scales for measuring objectives in preference models. In trying to capture this formulation knowledge in a program, the measurement properties of the scales will undoubtedly play an important role.

The **description** of an attribute conveys the interpretation of an attribute's value. Typically this will consist merely of textual definitions associated with the attribute name. If other URP symbols occur in the attribute definition (for example, to parameterize the scale specification), these are also described.

Combining Patterns

Once a substantial number of attributes have been defined in URP, it is useful to constrain the ways in which they may be combined in formulating a preference model. This type of specification serves to assist the user in selecting the URP attributes to include in the model, and may prevent some potential errors arising from uses of the attributes in ways unintended by the attribute definer. Combining patterns may also affect the structure of preferences over the chosen attributes.

Two mechanisms can be provided for specifying combining patterns. The first is to define *component relations.* A component relation is used to indicate that certain attributes are partial measures of other attributes. For example, if attribute Y_0 can be defined as a function of attributes $X_1, \ldots, X_k$, then the X_is are components of Y_0. In an URP formulation, Y_0 would automatically be treated as a vector attribute containing the X_is. These component relations may be built up into a hierarchical structure.

The second mechanism for specifying combining patterns is to define *inclusion links* between pairs of attributes. We can think of these links as specifying binary relations R and R', where $(X, Y) \in R$ implies that any preference model containing attribute X must also contain attribute Y. Similarly, $(X, Y) \in R'$ implies that any preference model containing X must *not* contain Y. R and R' are of course exclusive. Note that R' is symmetric, though R may not be.

8.3.2 Built-in Attributes for Medical Decision Making

Although medical decision problems present a wide variety of issues and criteria important to value judgments, it is possible and useful to define a few of the more common attributes and their utility-theoretic properties. Actual formulations for specific problems may use different combinations of these attributes, and may introduce others which have not been predefined. In this section, I will describe a few attributes which should be useful in a wide variety of medical decisions. Specification of the utility-theoretic properties of these attributes in URP's technical vocabulary is the subject of subsequent sections.

Life Years and Health Status

The first common attribute, illustrated in figure 8.1, is called *life-years*. As one would expect, the length of the patient's life is a major factor in most difficult medical decisions. Inclusion in a preference model, however, depends on stochastic relevance as well as importance. In a situation where the only difference between alternative strategies pertaining to lifetime is in short-term risk (operative mortality of a surgical procedure, for example), the appropriate attribute may be one that only distinguishes between surviving or not surviving beyond a certain time horizon. A program to help in the formulation task should have knowledge about relevance patterns such as this.

Another "generic" attribute is *health-status*. This attribute will denote different criteria in various kinds of medical decisions. In the simplest case—where there is only one relevant health criterion in addition to lifetime—*health-status* will simply represent that criterion. In the more general case, *health-status* will be a vector attribute consisting of several lower-level medical criteria, or the value of some *health status index* developed for specific kinds of problems. For reasons that will become apparent in section 8.6, the attribute *health-status* will only be used to describe states: conditions that exist at a point or over an interval of time.

Unlike *life-years*, *health-status* does not have a consistently defined measurement scale. For that reason, it is important to build in special attributes for the particular measures of health status dimensions that will be relevant in specific cases. These attributes would be connected to *health-status* through component relations.

An example of such a component attribute is the measure of physical function used by Torrance, Boyle, and Horwood [104] in their study of health state preferences. Their scale for the attribute *physical-function* is composed of six categories ranging from unrestricted mobility and physical activity to being dependent on others for getting around and lacking control of limbs. We can define this attribute directly in URP with little difficulty. The **scale-type** would be *ordinal* with a **scale** of (*integer-range* 1 6). The **unit** slot specifies the mapping from the integer scale values to the categories. In this example we would use the value one for the most restricted category of *physical-function* and six for the least. The description of the various categories can be taken directly from the English descriptions used by the scale's developers.

Other component attributes include those introduced in the example of section 2.4. There we might consider the attributes *disabling-stroke?*, *cabg-morbidity*, and *endart-morbidity* (X_2, X_3, and X_4) to collectively define *health-status*. To include these in the URP domain preference knowledge base we would need to generate attribute definitions similar to those described above.

Health Status Indices

Researchers in medical decision making and public health have compiled a considerable literature on health status indices and other mechanisms for measuring the desirability of health states.[2] While the indices have been developed for different purposes (health policy, clinical decisions, research evaluation) and under different methodologies (Gustafson et al. [44] compare indices developed using utility theory, actuarial approaches, and more ad hoc methods), they all provide a potentially useful scale for measuring difficult-to-quantify health status attributes.

Encoding these various measures as URP attributes is a feasible task that would have substantial value in bringing together a large body of research in a uniform framework. Provided with such a knowledge base, a health preference modeler would be able to pick and choose among a large variety of options in selecting attributes to include in the model. An additional advantage of predefined attributes related to health status indices is that these are often associated with well-tested measuring techniques (usually questionnaire responses or more objective criteria), and empirical validation over large groups of patients.

8.4 Predefined Qualitative Properties

It will often be possible to specify qualitative constraints on preferences for some of the built-in attributes for a particular application domain. These qualitative properties can sometimes be described directly in URP's terminology. Qualitative property assertions were described in section 3.2.1; a complete listing of the properties known to URP is given in appendix A.

8.4.1 Monotonicity

Many of these properties are quite simple and intuitive. For example, an obvious property of the attribute *life-years* is that its utility increases with the number of years. The URP assertion (**monotonic-increasing** *life-years*) means simply that

[2]The indices described by Parkerson et al. [81], Torrance, Boyle, and Horwood [104], and Fryback and Keeney [40] are just a sampling.

the patient will always prefer a longer life to a shorter one, all other things being equal.[3] Similarly, for most measures of health status, the assertion (**monotonic-increasing** *health-status*) will hold. Recall from table 8.1 that monotonicity assertions only make sense for certain types of measurement scales specified in the attribute definition. Constraints like this are useful to URP during model construction.

In many cases, it will be necessary to examine lower-level components of health status directly. These are very likely to have measurement scales consisting of discrete categories, as in the example of physical function described above. When these categories have some "natural" ordering (degree of physical functionality in the example) this ordering should be reflected by the index values. These natural orderings are often closely related to preference orders.

Preferences for *physical-function* are of course also monotonically increasing, but for other lower-level components of *health-status*, smaller values may be preferred, for example, required hospital visits per month. In still others, it is possible that there is an optimal point, say y_{normal}, which is preferred to values either above or below. To represent direction of preference for such an attribute, we would make two monotonicity assertions: (**monotonic-increasing** Y $[y_{lb}\ \ y_{normal}]$), and (**monotonic-decreasing** Y $[y_{normal}\ \ y_{ub}]$). Here y_{normal} may be a constant or may vary depending on other aspects of the problem.

Naturally, these properties are an artifact of the measurement scale chosen to represent the attribute. Scales can be inverted to change increasing preferences to decreasing ones and vice versa.[4] Often it may make sense to transform the non-monotonic attribute Y to another called Z, where $z = |y - y_{normal}|$. In this case, preferences over Z would be monotonically decreasing. In fact, the arbitrary nature of measurement scale helps to illustrate why qualitative properties from URP's technical vocabulary alone may not be intuitive. The interpretation of qualitative assertions depends critically on both the technical definition of the property and the meaning of the scale describing the attribute. Once again, the scale specification in the attribute definition is useful, because it tells us that a transform such as the one above results in a scale with a very different intuitive interpretation. Since the

[3]Of course, even this is a simplification. The possibility of situations so bad as to make one prefer a shorter life is examined in section 8.5, on independence relations.

[4]In fact, Torrance, Boyle, and Horwood [104] numbered the *physical-function* categories from one (best) to six (worst), rather than from six to one as I described previously. Using their indexing, utility for *physical-function* is decreasing.

transform from Y to Z is nonmonotonic, relations between magnitudes of Y are not invariant with respect to Z.

8.4.2 Risk Aversion

The dependence of intuitive interpretation on both measurement scale and technical definition is particularly important for the property of risk aversion. In purely mathematical terms, risk aversion corresponds directly to concavity of the utility function. To see how this technical definition translates to the notions conjured up by the property's name, we will have to look closer at the quantity measured by the utility attribute.

Bell [7] points out the importance of separating the different components of risk aversion. Several conditions may contribute to the risk aversion (concavity) of the utility function,[5] some of these with intuitive interpretations seemingly unrelated to "tendency to avoid uncertainty." For example, Bell [7] cites five influences on risk aversion for financial outcomes: non-linear measurement of the attribute, decreasing marginal value, effects of uncertainty on planning, risk anxiety, and decision regret. Although the legitimacy of regret and anxiety is questionable in a prescriptive framework,[6] the other factors may be applicable for a wide range of attributes. Gafni and Torrance [41] partition risk aversion into three components: a quantity effect (analogous to decreasing marginal value), a gambling effect (intended to capture the "attitude toward uncertainty"; not a very well-defined notion here), and a time-preference effect. Of course, the precise definitions of these factors depend on the particular attribute in question. As we will see below, some attributes may be associated with components of risk aversion in addition to these.

To accommodate the possibility of multiple factors affecting qualitative prop-

[5] And correspondingly, proneness or neutrality (convexity or linearity). To simplify further discussion, I will speak only of aversion and will assume that preferences for the attribute in question are increasing. Extension to the decreasing case or to risk proneness is straightforward.

[6] Unless the psychological factors behind regret and anxiety (such as disappointment, hopes, or fears) or the tangible effects on planning are explicitly treated in the model, there is no guarantee that preferences can be represented by a von Neumann-Morganstern utility function. Generalizations of expected utility (many of which are mentioned in a survey by Machina [72]) do not offer a solution, in my opinion, since even these usually require that all of the factors affecting hopes, fears, etc., are summarized in the statement of the prospect. Nevertheless, I believe that some of these models as well as psychological models of preference choice may turn out to be superior for descriptive purposes and, therefore, useful in the assessment process.

erties, URP must provide a mechanism for defining *influences*. An influence is an assertion that a particular factor tends to support or refute the applicability of a qualitative property. URP determines that a property holds if all of its non-neutral influences are positive. The problem of resolving qualitative influences is similar to that faced by qualitative reasoning programs (like those of Kuipers [67] and Forbus [39]).

Risk Aversion for *life-years*

Defining a qualitative property for a built-in attribute consists of specifying its influences and asserting the ones that are valid. The influences I have chosen to define risk aversion for *life-years* are *decreasing-marginal-value*, *time-preference*, and *age-preference*. This separation is far from obvious; the discussion below serves to define and delimit the concept that each influence represents.

Decreasing-marginal-value corresponds to the well-known phenomenon whereby agents derive smaller benefits from successive increments of the same commodity. The concept is particularly applicable for economic resources such as money (as in Bell's example noted above), since the agent is presumed to apply the resource to the most pressing needs or most value-gaining activities first. The value expected from each additional dollar decreases, since there are no items yielding greater value than the ones already purchased, neglecting effects of discrete purchases.

How does *decreasing-marginal-value* apply to *life-years*? Consider the collection of achievements and experiences a person may hope to amass in a long lifetime. Premature ending of that life would generally require that some of the items be removed from the collection—undoubtedly those of least importance. Progress toward goals may be viewed in a similar way. It makes most sense to take the steps yielding the most benefit, given that there is not enough time to complete the goal. In each case we reach the conclusion that additional years have decreasing marginal value.

Of course, this conclusion is dependent on several further assumptions. First, it is assumed that achievements, experiences, and progress toward goals is linearly related to additional years. This is at best an approximation, and it is easy to think of several reasons it should be significantly violated. Perhaps progress towards goals accelerates as they are approached, or one learns to achieve faster with experience.

The second major simplification concerns the continuity of these items. Measuring them in discrete milestones would be more realistic and would have a dramatic impact on risk aversion. We might still expect to see decreasing marginal value for life years in the long-term, but would also expect dramatically *increasing* value for lifetime increments in the immediate future. The usual example is the mother who would pay a large premium in life expectancy to achieve a greater probability of living through her child's college graduation. All-or-nothing goals present a similar situation.

A third problem is uncertainty. The decreasing marginal value argument works for money partly because agents know for certain how much they have to allocate towards consumption, at least in the short run. People do not generally have such reliable estimates of their lifetimes, so they cannot plan their actions optimally. This is perhaps a good argument for ordering all efforts and experience-quests in decreasing value, therefore contributing to risk aversion. Unfortunately, inherent sequentialities of life and other types of interactions may rule out this ordering in practice.

I have deliberately employed this narrow interpretation of *decreasing-marginal-value* to minimize overlap with the other influences on risk aversion for *life-years*.

These other influences are due to a peculiarity of the measurement scale for lifetime. Since *life-years* is a quantity of time, it is laid out in a distinct fashion: all additional time is added on the end. This may seem a trivial point, but it has a substantial (perhaps unexpected) effect on risk aversion.

To illustrate, let us consider two values of *life-years*: n and $n+k$. The difference is not just k years, it is *those k years from* $1985+n$ *to* $1985+n+k$. Higher values of n mean that the benefit from the extra k years of life will not be realized until further in the future. It is widely accepted that economic returns in the future have less value;[7] can this also be true for *life-years*? If so, we will say that *time-preference* is an influence on risk aversion.

Health economists and policy analysts make a habit of assuming positive time preference for *life-years*, usually by analogy to economic products. The validity of this analogy is questionable, since many of the justifications are specific to this class of goods.[8] Concern about legacy might explain the phenomenon, since earlier

[7] Though widely accepted, the rationale behind discounting can be quite complex. The presence of positive interest rates is clear evidence that discounting is appropriate, yet the underlying economic justifications combine several diverse components: value of capital for production, uncertainty, risk aversion, variation in income streams over lifetime, and irrational behavior are some of the explanations. Olson and Bailey [80] try to distill a notion of time preference from this conglomeration, but it is not clear that their results are transferable to health attributes—especially *life-years*.

[8] From the standpoint of *societal* preference for years of life the analogy may be more realistic. Raiffa, Schwartz, and Weinstein [86] point out that the discounting of life years in cost-effectiveness analysis

earnings have a greater contribution to the value of one's estate at a fixed point in time. Belief that the world will be a significantly better or worse place in the future can also have an impact on *time-preference*. There may be more compelling justifications, though none have occurred to me. My aim here is not to argue for or against the influence, only to suggest its potential applicability.

The final and possibly most important influence is *age-preference*. Each additional life year comes at a higher age, so preferences for years at different ages will naturally affect risk aversion properties. There are many obvious reasons for having different values for different ages, and some that are not so obvious. Physical attributes would tend to favor younger ages, though physical aspects taken account of in other health status attributes should not be considered here; the tendency of mental or emotional attributes are less clear. The criteria that would support influence in either direction are too numerous to discuss here. In any case, it is certainly an individual matter, one deserving serious attention in considering preference for *life-years*.

The purpose of this exercise has not been to decide whether individuals should be risk averse for *life-years*, but to point out the surprising complexity behind what was supposed to be an intuitive concept. It should be obvious at this point that the interpretation of the qualitative behavior of a utility function is highly dependent on the attribute in question, underscoring the need for a separate module for translating between domain concepts and technical modeling vocabularies.

Risk Aversion for Other Health Attributes

It would also be useful to define risk aversion properties for health-related attributes other than *life-years*. In creating these definitions, it would be necessary to proceed as above for each attribute, considering the possible influences on risk posture and their applicability in particular cases. Of course, we must also heed the requirements imposed by the measurement scales.

Consider, as a brief example, the attribute *hospital-confinement*, measured by the number of days spent in a hospital bed. Note that preference for *hospital-confinement* is **monotonic-decreasing**. Gafni and Torrance [41] suggest that preference in this case would also be **risk-averse**, since each successive day is considered worse as the patient becomes weary of her situation. If instead the patient

---

is justified under an assumption that the dollar value of a current life year is time invariant. While such an assumption and its conclusion are valid for making resource allocation policy, this does not necessarily have implications for the utility function of an individual patient.

became accustomed to hospitalization (therefore deriving less disutility each succeeding day), preference for *hospital-confinement* would be **risk-prone**.

8.4.3 Other Qualitative Properties

Similar definitions may be encoded for qualitative properties in addition to monotonicity and risk posture. These include the qualitative behaviors of single-attribute functions described previously, or other qualitative utility concepts not currently known to URP (such as descriptions of attribute dependencies, or multi-attribute risk properties).

8.5 Predefined Independence Conditions

In chapter 5 URP's extensive facilities for reasoning about independence conditions were described. To exploit this capability, the health preference knowledge base should contain as much knowledge as possible about the independence conditions that exist among the built-in URP attributes. Defining this part of the knowledge base is similar in spirit to the predefined qualitative properties; the difference is only in the nature of the assertions represented.

Consider as an example our two favorite attributes, *life-years* and *health-status*. Earlier (section 8.4.1) I claimed that longer lives are preferred to shorter ones, and represented that as a monotonicity assertion in URP. While this statement also includes the qualifier "all other things being equal," there was no qualification regarding the level at which the other attributes were fixed. To say that this level does not matter is equivalent to an assertion of **preferential-independence** of *life-years* from the other attributes (in this example only *health-status*).

But suppose that there is some value of *health-status* that is so bad that one would prefer to die than to live in that state of health. For example, an individual might express a desire that life-support be discontinued if she becomes a vegetable. We would encode this preference in the URP knowledge base by including the following two assertions in place of the original monotonicity assertion:

(**monotonic-increasing** *life-years* | *health-status* $\neq$ brain-dead)

(**monotonic-decreasing** *life-years* | *health-status* = brain-dead)[9]

[9] If the individual was merely indifferent to life expectancy as a vegetable, she would not care if the plug was pulled or not. A decreasing conditional utility function for life years implies that utility

Now that we have allowed a "state worse than death" we can no longer assume **preferential-independence**. Instead we can assert a weaker independence condition (introduced in section 6.11)

(**generalized-preferential-independence** {*life-years*} {*health-status*})

by noting that the preference ordering for *life-years* is completely reversed in the extremely morbid state.

These three assertions capture an important part of the individual's expressed preferences directly in URP's technical vocabulary. Just imagine how much smoother deciding medical ethics questions would be for the law courts if everyone recorded their preferences in URP assertions, rather than in casual remarks to family members!

Many of the other attributes that may be considered in health decisions can be uncontroversially declared to be independent (for some form of independence); these declarations should be included in the URP knowledge base to save time on utility formulations. For example, patients will not care much about cost for medical care that is paid through insurance, and society's preference for that money does not have any relation to the health of the patient it is spent on. Therefore, we can usually assert the strongest independence axiom for cost

(**value-independence** {*cost*} *everything-else*)

or at least assert it when we know that the expense is not borne by the patient. Preference for everything else is also independent of cost, since **value-independence** is a symmetric relation.

8.6 Special Functional Forms

The final component of a domain preference knowledge base is a representation for special-purpose functional forms that are considered useful for particular attributes or combinations of built-in attributes. Utility modelers in specific domains have developed and tested many such models, and it would be useful to incorporate these experiences in URP.

---

for immediate plug-pulling is strictly greater than that for delayed plug-pulling, which in turn is strictly preferred to living the maximum technological life. Note that these direction assertions say nothing further about the relative values of life under the different health states.

8.6.1 DEALE

The importance of the attribute *life-years* in medical decision making has been noted above. However, deriving a precise probability distribution over possible lifetime lengths is often difficult in practice. Beck, Kassirer, and Pauker [2] describe a simplified model of life years: the declining exponential approximation of life expectancy, or DEALE. The DEALE is computed solely from actuarial mortality rates (μ_{ASR}, the mortality rate adjusted for age, sex, and race), combined with excess mortality rates due to patient-specific disease conditions (the μ_is).

$$LE = \frac{1}{\mu_{ASR} + \sum_i \mu_i} \tag{8.1}$$

The DEALE is a very convenient model, since it is determined completely by the μs, which are relatively available in or computable from the literature and may be collected in different combinations for various patient states. Indeed, it is used routinely in decision analyses performed at NEMCH, including the example discussed in chapter 2.

LE in equation 8.1 may be considered as a proxy measure for *life-years*, or may be treated as a summary statistic for the distribution over survival times implied by the DEALE model. Note that LE is a *sufficient statistic*; there is exactly one DEALE survival curve corresponding to any particular LE. The interpretation decision is important, since consideration of the qualitative behavior for preferences over LE often depends on the particular point of view. In the example analysis of preference for DEALE life expectancy in section 2.2, LE was implicitly taken to be a proxy for *life-years*. It turns out that direction of preference (monotonicity assertions) and the presence of risk aversion for *life-years* correspond directly to monotonicity and risk aversion for DEALE LE under its interpretation as a summary statistic for a survival distribution. In general, transformations like this may not preserve these properties. It is important to define these correspondences in the knowledge base along with specifications for the special models.[10]

[10] In empiric tests of the DEALE approximation, Beck, Kassirer, and Pauker [2] only considered its validity as a measure of expectation—not as an approximation of the actual distribution of *life-years*. Therefore, extreme caution must be observed in taking the distribution literally, especially in consideration of preference properties of the attribute. This is a compelling argument for sticking to qualitative behaviors as much as possible, since few utility functions preserve implications over different interpretations of survival distributions (the risk neutral form is a notable exception).

8.6.2 Quality-Adjusted Life Years

One of the most widely employed utility scales of medical decision analysis is the quality-adjusted life year (QALY) [85]. A utility model based on QALYs is attractive to clinical decision analysts because it is computationally simple, relatively easy to assess, and it offers an intuitive interpretation for the numbers. The intuitive notion is based on the premise that a year spent in a morbid health state is "worth" some fraction of a year of good health. Pliskin, Shepard, and Weinstein [85] report necessary and sufficient utility-theoretic conditions for applying a utility function based on QALYs, which may be encoded directly in URP's technical vocabulary. I will outline these conditions below, following the development of Pliskin et al.

QALY may be appropriate when constructing a utility function for two attributes: life-years and health-status. It is assumed that health-status is constant over the entire lifetime or that we can choose a health state that would be equivalent to the particular time-stream of health states (denoted by a vector $\mathbf{q}$) we are faced with. That is, if $u_{q|y}$ is the utility function for health status given a particular number of life-years, y, then there exists a health state $\overline{q}$ such that $u_{q|y}(\overline{q}, \overline{q}, \ldots, \overline{q}) = u_{q|y}(\mathbf{q})$. For determining the applicability of QALY, we take the ability to choose the constant health state equivalent of a time stream of states for granted. Note that this task in itself may require a major preference modeling effort, and that the value of q may be a vector quantity made up of a variety of health-status sub-attributes.

The simplest version of the quality-adjustment utility function is

$$u(y, q) = yh(q) \tag{8.2}$$

where $h(q)$ is defined as the fraction of a year in the perfect health state that the patient considers equivalent to a year in state q. This form is valid under the following conditions:[11]

(**joint utility-independence** {*life-years*} {*health-status*})

(**constant-proportional-tradeoff** *life-years health-status*)

(**risk-neutral** *life-years*)

[11] The assumption that preference for life-years is **monotonic-increasing** is already part of the knowledge base.

Conditions one and three are familiar URP concepts, but **constant-proportional-tradeoff** is a qualitative property of attribute interaction, a category of utility-theoretic knowledge not yet known to the program. Assertions like these can be very useful for validating or constraining multiattribute forms, therefore future extensions of URP should exploit them extensively. Reasoning about these kinds of properties would not be very different from URP's manipulation of other utility concepts.

In cases where **risk-neutral** does not hold, the form of the QALY utility model is

$$u(y,q) = \begin{cases} \frac{1}{r}\left([yh(q)]^r - 1\right) + r & \text{for } r \neq 0 \\ \log\left[yh(q)\right] & \text{for } r = 0 \end{cases} \tag{8.3}$$

where r is a measure of risk aversion. This follows from **constant-proportional-risk-posture** for life-years, implied by the **constant-proportional-tradeoff** assumption. These results are derived in the paper of Pliskin, Shepard, and Weinstein [85]. I have restated them to illustrate how they may be incorporated in the URP knowledge base.

URP is already able to derive a two-dimensional form for $u(y,q)$ given **joint utility-independence**, and a single-attribute form for $u(y)$ under assumed behaviors **risk-neutral** or **constant-proportional-risk-posture**. The leap comes in the use of the function h, a nonlinear transformation of u_q, in equations 8.2 and 8.3. Thus, to implement QALY in URP we would have to give an interpretation for the function h—how it is assessed and its relation to u_q. More directly, we would support an assertion that QALY is appropriate at a high level, associating that assertion with the three (or two) implied assumptions listed above and the related functional form for life-years and health-status.

8.6.3 Time Streams

As illustrated in the utility model for health-status over a lifetime, it can be important to model preferences for outcomes represented by streams of states over time. Consequently, utility theorists have developed special models to deal with this kind of situation (see Meyer [78] for a discussion of the general issues). Due to the special structure of time-period attributes (sequentiality, constant dimension), intuitive concepts of preference may be stated succinctly—resulting in models far simpler than the general case. For example, for n-period streams of the form $x = (x_1, \ldots, x_n)$, the concept of *successive pairwise preferential independence* is equivalent to the URP concept:

(preferential-independence $(X_i\ X_{i+1})\ X_{\overline{i,i+1}}$**)**, $i = 1, \ldots, n-1$

To build expertise about preference modeling over time into URP, we would include definitions for those concepts developed specifically for utility analysis of outcome streams. In general, these concepts are combinations of general-purpose independence conditions, or other qualitative properties, that are already known to URP. Therefore, we can represent these in the knowledge base simply by using the concept name as a definition to be expanded into lower-level assertions (similar to URP's handling of *joint* independence described in section 6.9). Alternatively, URP can reason about these concepts at two levels: reduced to lowest terms and as top-level concepts. This latter possibility is advantageous when the concepts are closely related to specific utility models.

Several analysts have developed their own models of preference over time for specific application projects. We could include these models with their associated utility assumptions directly in the URP knowledge base, providing a wide variety of time models to choose from. For example, in an application of decision analysis to prostate cancer, Higgins et al. [49] employed an independence condition they called the "separability assumption" to simplify time streams of health states to an equivalent (with respect to preference) symptom-free outcome. Bell [3] gives sufficient conditions for a utility function that allows certain limited dependencies of preference among elements of a time stream. While URP can reason about the first principles from which both of these were derived, direct encoding of the special model leads to more efficient model generation and superior assessment, interpretation, and explanation ability.

8.6.4 Value of Life

Economists and public policy analysts have long studied models for placing economic values on life and health. Some of these models are close enough to utility-theoretic treatments that they may be encoded in the URP technical vocabulary for inclusion in a domain preference knowledge base.

For example, Howard's model for measuring the value of small risks of death or disability [53] assumes that consumption and lifetime can be traded off proportionately and that **constant-risk-aversion** holds over the function combining consumption and lifetime. From this he derives a measure of the value associated with small risks of death. The model is extended to disability under assumptions about health states similar to those required for the QALY model (described in section 8.6.2 above). It might be convenient to have this model, as well as other similar models, available in URP because of its simple structure and attractive explanatory power. Indeed, Holtzman's rule-based system for decision analysis in the domain of infertility problems [52] uses Howard's model as part of its underlying value function.

8.7 Summary: Using a Domain Preference Knowledge Base

The purpose of this chapter was to demonstrate that while a domain-independent mathematical modeling system like URP must reason in terms of a technical modeling vocabulary, it is possible to define a layer of domain knowledge that can be combined with the modeling system to yield a more complete modeling tool. In introducing the URP health preference knowledge base, I suggested that such a corpus may be useful as both an interface to expert systems and a library of model components for a human analyst. Advantages of separating the modeling kernel from concepts of the application domain are described in detail in another paper [111]; here I am more intérested in speculating about the content and form of knowledge about preferences in a particular domain.

The module that maps between domain concepts to the technical modeling vocabulary may in general consist of model templates, model component specifications, assertions about specific components, or special reasoning strategies. For URP, I have suggested that the forms of domain knowledge include:

- built-in attributes
- predefined qualitative properties for single-attribute preference
- predefined independence conditions among attributes
- special-purpose functional forms
- existing preference measures (e.g. health status indices) described in the literature

The health preference knowledge base is summarized in appendix F.

An expert system using URP for preference modeling would communicate with the modeler through the domain concept mapping module. Reasoning events in the program would be linked to preference concepts, directing the model construction process. For example, it would be up to the expert domain program to decide which attributes are important for a particular decision problem (for some expert systems this would be the same for every problem instance), and to determine which qualitative properties apply. We can easily envision a MYCIN-like system containing rules such as the ones illustrated in figure 8.2.

RULE001:
- IF There are no states worse than death
- THEN **(joint preferential-independence** {*life-years*} {*health-status*}**)**
  (monotonic-increasing *life-years***)**

RULE002:
- IF Financial costs are not borne by the patient
- THEN **(value-independence** {*cost*} *everything else***)**
  (monotonic-decreasing *cost***)**

Figure 8.2: Constructing preference models in a rule-based expert system.

Similarly, the domain expert system would use the result of URP's model analysis to make decisions, drive further preference modeling, and generate explanations and justifications. While it would be relatively straightforward to give a rule-based system access to URP's results (for example, by providing a predicate to test dominance, or describing the qualitative structure of the preference model), making intelligent use of the analyses is a subject for future work.

9. Conclusion

Having examined the mechanisms of preference modeling in some detail, it is appropriate to step back and examine again the task as a whole. In evaluating the contributions of this work from a larger perspective, it helps to keep the following questions in mind:

1. What are the problems that URP tries to solve?
2. How does URP's approach compare to related efforts?
3. What insights and specific achievements have evolved from this work?
4. What are the shortcomings of URP, and how might they be remedied?
5. What is the practical potential for including preference models in decision making programs?

In the discussions below I address these and other issues.

9.1 Perspectives on this Research

The goals and patterns of emphasis in this project have been influenced by interests in several different fields. I have found it useful to classify the potential benefits of this line of research into three areas, each representing a separate audience for this material. In this section I discuss what I believe are the important motivations and contributions of this work from each perspective.

9.1.1 Decision Making Programs

It seems an inescapable fact of technology that more and more decisions affecting society will be performed by computer programs, in one capacity or another. The spread of expert-systems-building methodology, including commercially available generation tools, has spawned many projects seeking to develop programs to make decisions in some specialized area requiring considerable expertise. These efforts have demonstrated enough promise to indicate that computers will play an increasingly large role in providing expert-level advice and decisions to broader categories of users.

Much has been written about requirements for these systems to become acceptable, particularly in sensitive areas like medicine.[1] Writers of these papers (Szolovits and Pauker [101], for example) often point out that programs must be able to explain and justify their conclusions and their knowledge bases should clearly reflect knowledge used (or usable) by experts in the field. But work on explanation has been mainly concerned with the elucidation of reasoning structures and mechanisms, providing justification for the program's inferences about the state of the world. At best, recommendations are justified by explaining the program's beliefs about what would happen under various strategies under consideration. There is never any explanation of why the program considers one outcome to be more desirable than another, because such comparisons are rarely (if ever) examined explicitly.

In using a decision-making program that employed preference models, the consultee would be able to query the system about its underlying preference assumptions. This capability is especially important if preferences can vary significantly among different decision makers.[2] Note that even a program that explicitly considers tradeoffs may be unexplainable if its conflict resolution method is sufficiently ad hoc.

Often, it is not necessary to have a sophisticated explicit preference model. In fact, most AIM (for *AI* in *M*edicine) programs that make decisions are in domains where a simplified objective structure is sufficient. For example, MYCIN [95] has a single objective—prescribe drugs to cover for all likely infecting organisms. Since most of the drugs in MYCIN's repertoire are fairly benign, this simple goal may produce results very similar to a program embodying the real goals of the patient. Here most of the effort is in determining the likelihoods of each organism—what might be called the probabilistic model formulation stage in a decision-analytic approach. In general, problems where the decision is uniquely determined by the result of the program's reasoning about the state of the world are not prime candidates for a

[1]Examples in this section are taken from work in knowledge-based systems for medicine. Much of the pioneering work in applied AI has been in medicine, and medical problems often have the characteristics of difficult decisions that motivate formal approaches. Besides, it is the area with which I am most familiar.

[2]Studies by McNeil, Weichselbaum, and Pauker [77] and Pauker, Pauker, and McNeil [83] illustrate the widespread differences in patient preferences for particular health outcomes, as well as the importance of these differences for decision making. A third paper [75] highlights variations in preference for *life-years* and argues that preferences are crucial in a large class of medical decisions. Kassirer [57] argues more generally that preferences of individual patients should play a central role in clinical decision making.

preference modeling approach.

Sometimes it is possible to deal with small deviations from the single-objective case without resorting to full-fledged preference models. ABET [16] chooses a therapy designed to restore a patient's proper electrolyte and pH levels. Conflicts between the individual electrolyte objectives are mediated through the use of an ad hoc urgency mechanism. Imbalances of different electrolytes may be compared in terms of health risk, in effect turning the multiattribute problem into one of a single attribute. Interactions which would invalidate this translation are treated in a special-case fashion. Note that this method of modeling objectives is entirely implicit in the program design, and it is doubtful that the designer thought of the model in these terms.

Many decisions in medicine as well as other fields, however, cannot be forced so easily into such a simple objective structure. AIM has generally avoided such problems, concentrating instead on diagnosis and reasoning about physiology. This is appropriate, since it represents the greatest part of medical expertise. And in cases where the diagnosis uniquely determines the decision, it is a complete solution. But for some important and difficult decisions, another type of expertise is required—a more fundamental decision making expertise. A necessary approach toward capturing that expertise is to explicitly represent and reason about the desirability of the various possible outcomes.

The argument for formal preference models is even more compelling for decisions made by computers than for human decisions. Those who object to the practical use of utility analysis for decision making can claim that unaided human decision makers may be taking into account criteria that are beyond our abilities to model. This claim does not hold that the decision making of humans is superior in every respect, only that utility theory alone does not (and fundamentally cannot) fully capture all of the subtleties of human preference choices. While decision analysts will argue that we do not want to imitate most of these subtleties, we are undoubtedly missing some positive features.

Computer programs that make decisions are themselves formal objects. For this reason, it is more difficult to argue that we would miss something by treating the preference structure formally. If we cannot tell what the preference model of a decision making program is by inspection and analysis, it is because the model is buried within the knowledge and control structures, not because there is anything magical or inherently unmodelable going on. The program designer has imposed an implicit preference structure on the process, which may or may not conform to her own internal model.

The idea of separate, explicit preference models is also a good policy from the standpoint of modularity in knowledge-based system design. Modifications or additions to the knowledge base may have unpredictable and undesirable effects on the program's objective structure if that structure is allowed to be implicitly distributed throughout the program. Once again, the use of ad hoc tradeoff mechanisms shares many of the defects of using no explicit mechanism at all.

In summary, the presence of an explicit objective structure enhances the capabilities of the system in three important ways. First, the decisions are more explainable and justifiable. Second, explicit representation of objectives greatly improves the modularity, and therefore the integrity, of the system. Finally (and perhaps most importantly), the decisions will be better—at least according to certain formal theories of decision making. Any program working within a reasonably complex decision space is almost certain to produce contradictions (with respect to axiomatic expected utility theory) if it does not expressly attempt to avoid them.

9.1.2 Utility Analysis

Researchers in decision analysis, as well as other areas of Operations Research and Management Science, often lament the fact that their techniques are not in widespread use by real-world decision makers.[3] This is certainly not because these decision makers do well enough on their own, or because the theories have not been developed enough. Bell [9] points out that decision analysts themselves fail to apply the methodology to their own problems. There may be many reasons for this failure, but the main cause is undoubtedly the tremendous effort and difficulty involved with using the formal models.

In the case of utility theory, there might be a question related to the admissibility of the axioms on which it is based. But this seems to be a negligible problem, since the axioms are generally very weak and acceptable to prospective decision makers. Besides, analysts are often extremely willing to use much cruder models that have either no theoretical basis or implicit assumptions that are demonstrably

[3] Little [70] begins a discussion of modeling tools (what would now be called Decision Support Systems) by declaring "The big problem with management science models is that managers practically never use them." While the statement is a bit dated, the sentiment is still quite true. The proliferation of personal computers and spreadsheet software have made the use of simple forecasting models more commonplace, but for the most part the more sophisticated models see little application. It is fair to say that with respect to theory, practice has been left in the dust.

inappropriate for the problem at hand. Another problem is a general skepticism regarding the legitimacy of techniques used to assess utility functions. Formally the theory is airtight, but in practice subjective judgments are an inextricable part of the process. Thus, the main barriers to the use of utility analysis are the practical difficulties of constructing and interpreting utility functions and a lack of confidence in the judgments required for their assessment. In the rest of this section, I will describe how URP and the overall framework described here may help to alleviate these problems.

Incomplete Specification of Preference Model

By removing the restriction that requires utility functions to be completely specified before analysis can begin, we add a great amount of flexibility to an assessment system. Even when employing the most simplifying assumptions, the task of assessing a utility function in complete detail is tedious, painful (requiring soul-wrenching decisions regarding unpleasant events), and subject to serious cognitive biases. If we are able to derive a decision before specification is complete, we alleviate much assessment effort. We may feel more confident in the result, because it depends on fewer and weaker assumptions. The inability to use partial information is probably the major reason that the mathematically complex models developed by utility theorists to represent weaker assumptions are rarely used in applications.

We have seen above that URP is able to derive some interesting conclusions and sometimes decisions from incomplete single-attribute utility functions. The example used earlier is far from unique; the commonality of situations where partially defined utility functions are adequate is the impetus for research in stochastic dominance. In a recent discussion of the important future developments in decision analysis, Winkler [114] emphasizes the promise of utility analysis tools that can take advantage of incomplete information.

In multiattribute utility, complexity, and correspondingly the strength of underlying assumptions, is measured by the degree of interdependence of preference among the attributes. By relaxing the restriction to complete specifications, we open up possibilities for applying multiattribute decompositions that admit limited kinds of dependence (another promising research area cited by Winkler [114]). Multiattribute utility assessment using current methodology is difficult enough in the simplest situation—to acknowledge attribute interdependence is tantamount to acknowledging that utility analysis is inappropriate. In fact, I am not aware of a real-world application that used anything more complex than a multiplicative utility function, or perhaps nested combinations of multiplicative utility functions. These functions are valid only under fairly strong independence assumptions. This represents the feasibility limit under current utility assessment methodology, which is clearly inadequate for credible preference modeling.

URP already has the facilities for generating decompositions based on more realistic assumptions. Combining this with the ability to reason about partial models (still in the future for the multiattribute case) should help to extend the feasibility frontier.

Reasoning About Underlying Assumptions

The validity of any utility representation depends on assumptions about the preference structure. At the most basic level, use of URP requires acceptance of axioms that imply the existence of a utility function.[4] Further conclusions about the utility function are based on additional assumptions. Clearly, these assumptions are important building blocks for model construction.

The underlying premises of an URP preference model (qualitative behaviors and independence axioms) are maintained as propositions in a truth maintenance system to provide for modular assertion and retraction. At the very least, modelers who use this program for developing a utility function will be aware of what underlying assumptions they are making. This may induce them to think more carefully about their simplifications. Further, examiners of the model can then see what presuppositions went into the formulation, and use that knowledge to decide whether to believe it or not. The whole process becomes more transparent and more honest.

Another advantage is the explicit representation of the relations between various types of assumptions. This knowledge is built into the program, to be used for both reasoning and explanation. Modelers may find that a particularly useful assumption is implied by several different sets of weaker primitive axioms and can choose the set which appears most justifiable. It may also prevent the analyst from making strong assumptions (or thinking that she is making strong assumptions) that are unnecessary for the model in question.

Further discussion about the importance of assumptions in a more general mathematical modeling context may be found in a forthcoming paper [111].

[4]On startup, URP asserts the primitive proposition "**(von-neumann-morganstern-axioms)**." If this assertion is ever retracted, then URP will do something drastic like erase all conclusions and halt. Except for this and some assumptions like continuity and differentiability employed by the mathematical reasoning system, all premises are explicitly asserted by the user.

Structural Sensitivity Analysis

Analysts will often attempt to validate their results by performing a sensitivity analysis on their model. The purpose of this analysis is to examine the effect of perturbations in parameter values on the overall result. If it turns out that small changes of the parameters over their realistic ranges does not have a large effect—that is, it does not change the decision—then the analyst will be confident in the correctness of the result.

It may be argued that the restriction to completely specified models is not much of a restriction, since we can simply choose baseline values for the parameters arbitrarily and use sensitivity analysis to account for variations. But this will not lead to a satisfactory validation, because only a few parameters are varied at a time. In moderately complex decision analyses, there may be dozens of parameters. In such cases, the model may prove unjustifiably insensitive to variations in only a few. Variation of more than two or three parameters turns out to be both computationally infeasible and difficult to interpret.

One reason that the model may be unjustifiably insensitive is that there may be couplings among the parameters that are not explicitly represented. That is, a built-in redundancy or other interaction captured in the estimates. Part of this may be eliminated by decomposing the higher-level parameters of the decision to a smaller number of common "influences" [53] affecting them, and expressing the higher-level parameters as functions of the lower-level ones (this mechanism is available in decision analysis tools). But this does not solve the problem, since there may be obvious relations among the parameters without any easily identifiable common components. Choice of parameters is often dictated by external factors such as measurability, convention, or intuitive appeal, which may make them difficult to decompose. Perhaps these may be captured by some other sorts of constraints, using inequalities rather than definitions. Something like this may be possible even within the current framework for sensitivity analysis, but it is more natural in a system that avoids baseline estimates in the first place.

A fundamental limitation of traditional sensitivity analysis is that we can vary the parameters, but not the underlying model. It is impossible to ask questions like "What if this particular assumption is wrong?" Recognizing the importance of such questions, Farquhar [30] has called for the development of procedures for performing ***structural sensitivity analysis*** which would examine the effects of more fundamental changes in the model.

The potential for structural sensitivity analysis in URP has been pointed out earlier in this thesis. In the multiattribute decomposition example (section 2.4), retraction of an assumption led to a significant restructuring of the preference model. Assertion and retraction of qualitative behaviors may be handled dynamically, resulting in a wide range of functional forms and parametric constraints. The possibility for switching between alternate psychological theories for interpreting preference choices was discussed in section 7.4.2. This would be a particularly novel form of structural sensitivity analysis.

Efficiency in structural sensitivity analysis is achieved through maintaining dependencies between premises and derived inferences about the preference model. At the propositional level these dependencies are recorded by RUP. Dependency-directed reasoning has been a subject of much study by AI researchers in recent years; it would be no surprise if a capability for structural sensitivity analysis was first developed in systems employing AI techniques. It will be interesting to see how dependency-directed inference may be incorporated into other aspects of URP's reasoning about preference models. McDermott [74] offers suggestions for maintaining dependencies among inequality assertions. His techniques are likely to be useful in extending URP to handle preference choice information—basically assertions of inequalities among utility function evaluations.

Explainability

It is undoubtedly easier to describe in intuitive terms a preference model that is based on qualitative characteristics than one that is a purely mathematical object. Each conclusion made in constructing the model is traceable to base-level premises, which should have some meaning to whoever asserted them (a human user or some level of interface). The program should be capable of providing a good explanation of the model structure—after all, the program constructed the model in the first place. This is somewhat analogous to the explanation work of Swartout [99], where the program is more explainable because it is written by an automatic programmer. In our case, the model is more explainable because it is constructed by an automatic modeler.

But of course, explainability does not come for free. As Clancey points out [19], the knowledge that is best for reasoning is not necessarily appropriate for explanation and justification. And Swartout's system [99] would not work with just any automatic programmer. XPLAIN generates the expert system based on a detailed domain model and a knowledge base of domain principles. Special explanation problems arise in mathematical modeling applications, particularly in describing quantitative relationships. Kosy and Wise [65] have demonstrated an approach for explaining simple spreadsheet models; it may be possible to extend their methods to other quantitative relationships. The major explanation issues have not been explored extensively in URP, but there is reason to suspect that there will be some improvement over traditional approaches. Decision-analytic models which do not maintain ties between the quantitative constructs and their qualitative justifications are notoriously opaque.

Utility Analysis: Summary

Some of the benefits of the URP approach for utility analysis were not discussed in this section, because they are described in some detail elsewhere in the thesis:

- Encoding knowledge about utility theory (specifically multiattribute decomposition) in an explicit representation (chapter 6)
- Interpretation-independent representation of assertions (sections 3.2.2 and 7.1)
- Tying model assertions to domain knowledge (chapter 8)

In summary, facilities provided by URP and potential capabilities of future programs built within the URP framework address many of the problems associated with utility analysis under current methodology. By emphasizing underlying assumptions and allowing partially specified functions, URP supports a qualitative view of utility. This qualitative basis in conjunction with URP's framework for dominance-proving and structural sensitivity analysis provides a response, or even a solution to each of the complaints about the multiattribute utility approach enumerated by Starr and Zeleny [97] in a recent survey of decision making with multiple criteria.

9.1.3 Mathematical Modeling

The third and final perspective comes from viewing utility analysis as just one kind of mathematical modeling. Mathematical models are used extensively in just about all branches of physical and social science, and one might regard the skill and knowledge involved in creating and analyzing them an important type of expertise. Of course, this expertise is very different for different modeling tasks, but undoubtedly there are some general principles and mathematical capabilities that are broadly applicable.

I have found the preference modeling task to be a particularly good example for studying mathematical modeling expertise, because it deals with many of the central issues I believe arise more generally. In addition to the issues of reasoning with mathematics, there are a set of issues pertaining more specifically to the construction of formal structures representing a simplification of the real world. The remainder of this section highlights the features of URP which may have implications for general issues in building knowledge-based systems for mathematical modeling.

Mathematical Expertise

A program that is expert at mathematical modeling should naturally be competent in mathematical reasoning. It is important to note the distinction between numerical computation and mathematical analysis. We are interested here in the latter, deeper type of mathematical capability.

URP's mechanisms for reasoning about qualitative behaviors represent a step in the direction of deeper mathematical knowledge. Because the program reasons about behaviors in terms of their mathematical definitions, the concept associated with them may be applied in a variety of ways, some unanticipated during the original design of the program.

Usually mathematical models are constructed to answer a set of questions that may be expressed in a formal language. If the types of questions are known in advance or are of a highly restricted form, it is feasible to develop algorithms to answer each type of allowable question for each qualitatively distinct possible model structure. But there is often a substantial benefit to be gained by taking a more general approach. We may wish to restructure the models dynamically (as in structural sensitivity analysis, discussed above), or permit a much broader range of queries about the model. In such cases, the strict algorithmic approach breaks down, requiring a much more flexible kind of problem-solving behavior. To gain this flexibility, expert programs for mathematical modeling will have to look a lot more like theorem provers than like the model analysis packages of today.

Emphasis on Assumptions

I have already discussed in some detail the benefits of reasoning directly about assumptions underlying preference models. The only thing to add here is the observation that assumptions are a fundamental part of virtually all modeling disciplines, therefore, we should expect analogous benefits in the general case.

Measurement Properties

As argued in section 8.3.1, programs that manipulate quantities should know something about the measurement properties of those quantities. This can prevent some "common sense" errors as well as indicate allowable operations and transformations that may help in mathematical problem solving.

Technical Vocabularies

The notion of a technical vocabulary developed in section 8.1.2 is a useful concept in the design of knowledge-based systems for mathematical modeling. The vocabulary is the language of the domain-independent modeling system, which ultimately defines the scope of the modeler's expertise. This language also provides a basis for building knowledge bases specific to modeling in particular domains, as illustrated by the health preference knowledge base specified for URP. In general, a well-designed technical vocabulary will lead to a clean separation of modeling and application-area expertise.

Flexible Use of Algorithms

When it is necessary to use algorithms coded directly in a general-purpose programming language, it helps to provide the overall modeling system with some knowledge about the algorithms available. Though the ideas have not been developed very far, URP's facility for specifying dominance-testing algorithms (section 4.3) illustrates some kinds of information that should be useful. As mentioned in section 9.2.2 above, several other programs, particularly in statistics, have provided similar facilities for specifying analysis algorithms.

Other Components of Modeling Expertise

Work on URP has scratched only the surface of the mathematical modeling issues described above. In addition to these, there are several kinds of knowledge not used by URP that are in general important for programs that reason about mathematical models:

- Empirical verification of assumption validity
- Heuristics to direct search for valid model structures
- Effects of approximations

- Computational complexity of model analysis algorithms
- Qualitative interpretation of model results

The possibility of including these components of modeling expertise in knowledge-based systems has not been adequately explored. All are probably necessary for a truly expert automatic modeler.

9.2 Other Work

9.2.1 AI and Decision Analysis

Considering the large amount of research that has been concerned with computerized decision making and decision aiding, it is surprising (and regrettable) that there have been so few attempts to integrate approaches from artificial intelligence (AI) and decision analysis (DA).[5] I believe that this lack is due to the prevailing perception of the fields as competitive—that it is necessary to choose one or the other as the "better" methodology. Of course, neither strategy is clearly "more right," rather each has particular strengths and weaknesses which are highly dependent on the situation. Recognizing this, some researchers have attempted to identify the features or stages of decision situations in which either methodology is more or less appropriate. An example is the discussion by Szolovits and Pauker [100] of issues arising in trying to cope with uncertainty in medical diagnostic programs. While a method of deriving beliefs does not necessarily imply a method of choosing actions based on beliefs, many of the issues they discuss are relevant to the choice between AI and DA as an overall approach.

A different kind of question to ask is "How might ideas from either discipline complement the other?" There are many conceivable ways in which AI techniques could enhance DA tools (the AI→DA direction) and DA approaches could be incorporated in AI programs (DA→AI). Although some have recognized the potential of integrating the two methodologies [32], serious attempts at this synthesis have been few and far between. The technical issues involved have not been adequately explored, and the precise benefits to be gained from the integrated system have not been clearly formulated.

[5] The use of probabilistic methods in an AI program does not constitute an integration of DA into a system. At the very least there must be some principle of expected utility maximization as a choice criterion. Bayesian or pseudo-probabilistic inference schemes are for managing beliefs, not for making decisions based on beliefs.

The rest of this section presents some details of a few early attempts to exploit AI techniques for the improvement of DA systems. Though none of these efforts are devoted primarily to utility analysis (and I know of no other work that is), a review of these systems serves to develop a perspective for the research described in this thesis. Use of URP for decision analysis should be considered in the context of these integrated approaches.

Decision Tree Generation

Structuring a decision problem in the form of a decision tree is one of the primary tasks in performing an analysis. One of the more promising avenues for an AI→DA approach is to use a domain knowledge base to formulate a decision problem from a description of the situation, for example, disease states of the patient.

Rutherford et al. [88] describe a decision analysis program for Hodgkins disease which employs some novel methods for representing knowledge for the generation and evaluation of decision trees. Rather than directly encoding the tree as a structure of nodes with associated probability and utility expressions, the Hodgkins program uses predefined concepts of tests and treatments to dynamically construct a decision tree at decision evaluation time. The structure of the tree is compiled into the interpreter, in the sense that the decision at any point is a choice among the remaining tests and the available treatments. The effect of tests is to revise the likelihood of the various patient conditions, perhaps exposing the patient to some mortality risk. Treatments are always terminal nodes of the decision tree. The expected utility of a treatment is calculated directly from the input distribution of patient conditions.

Even though this simple decision structure is compiled into the program, the system's use of test and treatment as high-level concepts provides a flexibility not found in typical DA tools. Characteristics of the patient may be used to determine the list of available tests and treatments, or these may be controlled directly by the user. It is even possible for the user to define her own tests and/or treatments, to modify existing ones, and to specify new strategies or parts of strategies. Since any of these changes results in widespread structural modifications of the decision tree, such a task would be extremely cumbersome using traditional "tree-defining" DA programs. Algorithms designed specifically for optimizing test selection sometimes avoid these difficulties.

Hollenberg's Decision Tree Builder (DTB) [50] generates decision trees using a medical knowledge base of diseases and interventions (tests and treatments). Unlike the Hodgkins program, DTB is intended to handle a broad range of medical decision problems. Consequently, its representations are considerably more general, and its tree generation correspondingly more flexible. A disease may be parameterized by *attributes*, which in turn may influence the applicability of various interventions as well as probabilities and utilities in the model. Tests and treatments may indicate or modify the values of disease attributes. A simple control structure directs tree construction, employing a model of patient states for bookkeeping purposes.

Hollenberg's work has also explored the use of *influence diagrams*, a representation for decision structures developed by Howard and Matheson [54]. Influence diagrams offer several advantages over decision trees as an internal representation for a decision formulation in automatic DA model-builders. Like decision trees, influence diagrams are graphs where the nodes represent decisions or chance events. The links in influence diagrams represent information and probabilistic dependencies among the nodes. Because decision trees must be strictly hierarchical, they contain a substantial duplication of structure that is unnecessary in influence diagrams. In fact, the number of nodes in a decision tree grows exponentially with the size of an influence diagram. Furthermore, a decision tree representation cannot take advantage of known independence relations, since a left-to-right ordering is always imposed on events. Though sometimes this ordering reflects the temporal course of the situation, often the precedence choice is quite arbitrary.

Consultation Systems

The systems described thus far, including URP, would be characterized as primarily AI→DA approaches. Attempts to integrate AI and DA in systems for direct consultation, on the other hand, contain elements of both directions.

RACHEL, designed by Holtzman [52], is a program to aid in decisions faced by infertile couples. The program has a prespecified formulation of the most general form of decision problem, represented as an influence diagram. For consultation, RACHEL uses domain knowledge encoded as production rules to determine which parts of the internal model should be instantiated for particular cases. In establishing and parameterizing the model, the program is controlled by a backward-chaining inference engine. Parameters of the model are filled in by user input or inference rules which may take the structure of the model into account.

"Intelligent decision systems" built along the lines of RACHEL incorporate domain knowledge to cover a well-defined class of decisions (infertility in this case, end-stage renal disease in a precursor system also developed by Holtzman [51]).

The domain knowledge is primarily in the form of a general decision model, encoded as an influence diagram with assessment functions for each node of the graph. Constructing a model for a particular decision is largely a matter of reducing the template model that is built into the program. Note the contrast between this and the tree generation approach described above, where models are constructed by combining primitive components in a controlled fashion.

Lehner and Donnell [69] propose an architecture for a decision aiding system based on a combination of DA models and rule-based expert systems. Basically, their system consists of a prespecified utility model that is parameterized by the application of inference rules. Although this approach is severely limited, it represents a simple kind of integration that can be routinely implemented with current application technology.

It would be interesting to identify the domain-independent component of modeling expertise and separate it from knowledge specific to the decision-class supported by the consultation system. Such a clean division is not apparent in existing approaches.

9.2.2 Knowledge-Based Approaches to Mathematical Modeling

At a recent major AI conference (IJCAI 1983), four papers were presented that described projects merging expert systems and mathematical modeling. Though differing widely in application area (oil-well data interpretation [15], portfolio selection [20], flood control [21], and macroeconomics [87]) as well as in the mathematical modeling discipline employed, each takes a similar approach toward integration: applying production rules to parameterize model templates. This sample is representative of the existing expert systems that incorporate a mathematical modeling component.[6]

A rule-based system that fills in parameters often makes mathematical models easier to use. An extra layer is placed between the model and the user, translating qualitative criteria to the numeric input required by the model analysis package. Or perhaps the translation is in the other direction, from the model's numeric output to qualitative concepts more intuitive to the user. In either case, the rule base is useful for connecting mathematical modeling to other sorts of automated reasoning

[6] I am excluding from this discussion expert systems that perform tasks (such as scheduling or inventory control) in areas where mathematical models are traditionally employed.

and certainly provides an improvement over unaided use of modeling algorithms. Some of the issues in developing these interface layers are discussed by Weiss et al. [110].

Nevertheless, systems that treat the modeling algorithm as a black box are unduly limited in the range of mathematical models they can construct and the possible conclusions they can derive from the analysis. This rigidity typically restricts uses of the program's knowledge to modeling in only one narrow application domain or even to a single problem instance. For example, the portfolio selection program built by Cohen and Lieberman (FOLIO) [20] contains a template goal programming formulation which it parameterizes and then solves by calling the black-box algorithm. Since there is no domain-independent knowledge about goal programming, it cannot adapt to different model *structures* that may be appropriate for specific cases. While sometimes it is possible to mimic a structure by setting parameters of an overly general model appropriately, the algorithm cannot take advantage of potential simplifications. A more serious problem is that the program is unable to deal with violations of the assumptions underlying the model. (The model in FOLIO is equivalent to an additive-linear multiattribute utility function—a form that is valid under only the strongest of independence conditions.) A program with a more general modeling expertise might be able to handle cases where some of the default structural assumptions are inappropriate.

One important component of modeling expertise is to choose among alternative analysis algorithms. An expert system built specifically for this purpose is SACON [12], a program that advises engineers in the use of a structural analysis package. The package is a large collection of programs for specific modeling tasks. SACON uses knowledge about the package to determine which of its algorithms should be used to analyze the structure that the engineer is interested in. Knowledge about algorithms includes items like error tolerance and the types of structural phenomena that are considered (fatigue, for example). This knowledge is encoded in production rules. URP makes use of the same types of knowledge about dominance-testing algorithms but records the information in a schematic data structure. Similar specifications for algorithms have been employed in knowledge-based approaches to statistics (Blum's RX [14] and a regression system developed by Gale and Pregibon [42]).

Other expert systems perform some kinds of quantitative reasoning. Kunz's AI/MM [68] utilizes simple quantitative relations between physiological parameters within a rule-based system. The mathematical modeling components of these systems are not nearly as sophisticated (that is, complex to apply and interpret) as those we are most concerned with here, but the development of techniques for coupling categorical and quantitative reasoning methods is potentially useful for our purposes.

9.3 Limitations

Deficiencies of URP have been pointed out throughout this thesis report. Many of these are due to time constraints; it was simply impossible within the scope of this project to explore every issue of the preference modeling problem that the approach described here might address. Of course, the implementation of even the best-developed parts of the task leaves much to be desired. Other limitations are more fundamental in that it is difficult to foresee their solution within the currently envisioned preference modeling paradigm. The sections below examine the weak points in more detail, suggesting possibilities for improvement in some cases.

9.3.1 Preference Modeling Issues

While URP is useful for a substantial part of the preference modeling task, it does not quite cover the high-level view depicted in figure 1.1 on page 5. There is at least one complete path from input assertions to a decision, though many of the paths we would like to take using the system run into missing components along the way.

Chapters 7 and 8 discuss how the issues of assessment and domain preference knowledge would be addressed within the URP framework. The facilities described there have not been implemented, leaving URP with several gaps in its preference modeling ability. Until an assessment facility is implemented, the proposition that we can perform structural sensitivity analysis and interpretation under descriptive theories of choice must be regarded as an untested hypothesis, although I think there is a strong case for believing that such capabilities can be achieved. It is difficult to argue forcefully that URP will be easy to use by expert systems or human analysts without providing the technical-vocabulary/domain-concept mapping supported by the domain preference knowledge base.

Another large gap in URP is its lack of support for asserting qualitative behaviors of multiattribute utility functions and lack of any mathematical reasoning about multiattribute forms. In contrast to independence conditions, multiattribute qualitative behaviors describe the nature of *dependencies* among attributes in a utility function. Like single-attribute behaviors, they are useful for characterizing mathematical properties of the function as well as indicating the validity of dominance-testing procedures. While I do not anticipate any difficulties in defining these behaviors and performing logical inference based on them, a significant extension of QM would be required to represent and reason about multivariate functions. Nevertheless, an extended QM would undoubtedly be able to handle many of the kinds of inferences that would be useful in multiattribute problems. Qualitative behaviors of these functions expressible as properties of partial derivatives (a large class of the interesting behaviors) can be handled using QM's single-variable representation.

There are as yet no general-purpose dominance-proving or constraint reasoning procedures to address the problems of chapters 4 and 7. While URP should make use of its special-purpose dominance knowledge as much as possible, it should be nevertheless be somewhat competent at employing weaker methods in the more general cases. Similarly, it will be necessary to choose appropriate constraint manipulation methods to implement the flexible assessment capability discussed above.

9.3.2 The Implementation

Even the parts of URP that have been fully implemented could use quite a bit of revising, tuning, extending, and fixing. Because this was an experimental project, pieces of the program were only developed to the point that they could demonstrate that the desired capability was achievable. The major thrusts of improvements to URP's implementation should be toward efficient use of existing mechanisms, an expanded knowledge base, and the development of user interfaces.

The most pressing efficiency consideration is in the derivation of multiattribute decompositions. There is considerable opportunity for optimization of the lower-level algorithms involved and for improvements to the simple control structure currently in use. Perhaps the most promising avenue for efficiency gains is in the heuristic control mechanism suggested in section 5.4.3.

The knowledge base might be expanded in several directions: to include more behaviors, more independence axioms and theorems, and more stochastic dominance algorithms. Of these, the last is probably most important; URP currently has a very limited repertoire of dominance-testing procedures. Though investigations are preliminary, it appears that there is quite a large body of useful results to draw from [1].

The existing program is a perfect example of "user-hostile" software. Since the mechanisms were built without any precommitment to a particular mode of usage, it is generally not possible to interact with URP without knowing something about the inner workings of the program. This situation will clearly have to be remedied before the system can be useful as a utility analysis tool.

9.3.3 The Approach

Most troublesome are the limitations of preference modeling with URP that do not appear to be just a matter of gaps in the implementation. This section identifies a few of the more fundamental difficulties which I do not have any particularly good ideas about resolving.

Driving Model Construction

The first problem concerns URP's failure to cover the full preference modeling task as diagrammed in figure 1.1. Recall that in the figure there is a control link tying the results of model analysis to goals of model construction. Unfortunately, URP does not actually close that loop, and it is not obvious to me how to do so. While there are some fairly simple tactics which might be applied when analysis fails to derive a decision, such as establishing preconditions for a more powerful dominance routine or finding an incrementally simpler utility expression or decomposition, these do not take advantage of information derived from the failed analysis.

Complexity

Probably the most apparent problem of the entire approach is that URP is certainly not a "simple and elegant" solution to the preference modeling problem. It is comprised of many complex mechanisms and relies on some fairly sizable reasoning tools (RUP and QM, which in turn uses MACSYMA) to do some of its work. Extensions mentioned above would require even more complexity and additional high-powered tools.

Part of this complexity is a conscious response to the overly simplified view of preferences typically taken by decision-making programs. One of the major points of this thesis project was to demonstrate that it is possible and desirable to exploit the large body of theory concerning formal preference models. Given the present structure of utility theory, this requires a lot of separate, complicated mechanisms. It may be that some of these mechanisms will turn out to be more useful than others—the fact that they are separate also implies that there should be some benefit to each mechanism in isolation. Thus, I expect that further work will concentrate on identifying the pieces that are most important in particular preference modeling tasks. These mechanisms could then be tuned to provide the best performance for those tasks.

Formulation

It is an unfortunate fact that URP's *model construction* starts with considerably more than scratch. A large part of the expertise of a utility analyst is in choosing the right attributes to model. Since URP abdicates this responsibility, any success it has might be attributed to having fed it the right formulation to begin with.

This is a difficult problem, and I am not optimistic that it can be easily solved. Choosing attributes requires a great amount of domain and real-world knowledge of the sort that cannot be contained in a general-purpose preference modeling program. Some of this knowledge may be supplied by a domain preference knowledge base of the sort outlined in chapter 8, though the actual methods for using it (particularly for attribute selection) have not been worked out. There are, in addition, some domain-independent principles which may help to determine the applicability of a set of attributes for a given problem [59] [60]. Defining measurement properties (discussed in section 8.3.1) may help. Formalizing these principles is a challenging though worthwhile exercise, crucial to the development of a truly useful utility analysis system.

9.4 Future Work: Planning with Preferences

Much has been said here about the potential for preference modeling in decision making programs without any serious discussion of how it might be incorporated in actual program designs. URP is precisely what its acronym suggests: a collection of programs which perform various sorts of reasoning about utility. Embedding this reasoning in a larger decision making system architecture would require a considerable engineering effort. And undoubtedly the issues involved in an integration of preference modeling in AI programs would vary greatly across different problem-solving paradigms.

To test the usefulness of some of the ideas developed in this project, I will attempt to incorporate elements of preference reasoning in a domain-independent planner and apply the program to problems in patient management. Planning is a particularly good framework for studying the implications of preferences, since constructing a plan requires many choices among alternative actions. It is no coincidence that planning spaces bear a close resemblance to decision trees.

Charniak and McDermott [18] (page 524) point out that little AI planning work has incorporated a decision-theoretic basis for choice.[7] Indeed, the problem of choice has received minimal attention of any kind. State-of-the-art planners have no means to reason about partial or uncertain satisfaction of goals, much less to explicitly consider tradeoffs in choosing planning steps. Developing such a capability would entail an expanded language for describing objectives, a means for representing uncertainty, and mechanisms for expressing and reasoning about the desirability of outcomes.

From the point of view of the proposed planner, URP is a vehicle for exploring this last issue. Experiments with URP should help to identify the most useful types of assertions about preferences and how to reason about them. The goal will be to select a subset of the conceivable URP assertions to include as primitives in the planning system. My expectation is that most of the qualitative behaviors and independence conditions already defined in URP will turn out to be useful, in addition to qualitative multiattribute properties not yet implemented.

Work in planning will also highlight an issue that has not been explored much with URP: how these preference assertions may be used in *formulation*. Knowledge about the decision maker's objective structure will often be useful for determining whether a tradeoff situation exists and which are the pivotal attributes of the situation that must be included in the model.

Any more detailed discussion of how preference modeling in the URP framework would be accomplished in a planning environment would be purely speculative at this stage. In any case, the ability for formulation (or even simply recognition) of tradeoff situations alone would be a substantial advance in AI planning technology.

[7] Feldman and Sproull [31] argue for the use of decision theory in robot planning. Though their numerical approach to utility shares many of the shortcomings of heuristic evaluation functions, using outcome desirability to plan actions is shown to be valuable for revealing tradeoffs, allocating planning effort, and directing information acquisition. As far as I know, their ideas have not been incorporated in an actual planning system.

9.5 Contributions

9.5.1 Summary

In my opinion, the important contribution of this work is in its overall approach toward preference modeling, rather than in any of the particular capabilities that have been described. Nevertheless, in the course of attempting to demonstrate the potential advantages of a qualitative view of preferences and a knowledge-based representation of utility theory, several more specific goals were accomplished. These achievements have all been chronicled in the thesis body; the remainder of the section is a brief outline of what I believe to be the most important contributions.

Qualitative Behaviors

- Explicit maintenance of the qualitative premises underlying single-attribute utility functions
- Integration of logical and mathematical relations among behavior assertions and function objects
- Adaptation of QM to handle some types of incompletely specified functions

Stochastic Dominance

- Knowledge-based approach tying dominance tests to qualitative preconditions for their applicability
- Framework for incorporating algorithms in the overall system

Multiattribute Decomposition

- Language for describing independence axioms and decomposition theorems, with interpreter/theorem prover
- Encoding a substantial body of multiattribute utility theory in this language, covering the major results of the field
- Generation of hierarchical decompositions without prestructuring by the user

Assessment

- Framework for model-independent use of hypothetical preference choices
- Potential for interpretation under descriptive theories of preference choice

Domain Preferences

- Notion of a technical vocabulary for domain-independent modeling systems; nature of the mapping to domain concepts
- Using properties of measurement scale to capture some "common-sense" features of a quantity
- Construction of a fragmentary health preference knowledge base

9.5.2 Outlook on Preference Modeling

Of greater interest than the merits of the individual contributions themselves is the potential for integrating them into a coherent and comprehensive preference modeling system. While there is quite a large disparity between URP's capabilities and this ultimate objective, there are several reasons to believe that the goal is a realistic one. From the vantage of the existing URP implementation, there are some obvious extensions which could be undertaken to cover a larger share of the overall task.

Only a few of the limitations noted in section 9.3 seem to be fundamental. Most would require substantial engineering effort, but should be possible with off-the-shelf AI technology. The overall design is quite complex, but simplifications and generalizations are likely to become apparent as more of utility theory is formalized. Indeed, experience in implementing URP indicates that there may be considerable "hidden" generality that can be exploited by an automated reasoner. Without a program that can flexibly apply the theory, orthogonality of utility concepts does not get much attention in the literature. For example, two generalities of utility theory that do not often get applied in practical model structuring (though they are exploited in proving decomposition theorems) are the ability to treat sets of attributes as single vector attributes and the extension of independence relations to analogous relations with fewer conditioning attributes (URP theorem 23). Applying these together with knowledge of two-attribute functional forms, URP automatically generates hierarchical decompositions. This opportunity "falls out" of the structure of the theory; it is not practically important without an automatic model constructor.

We should also expect that the comprehensive modeling system will become more feasible with the advancement of related AI technologies. Work in qualitative mathematics should result in more efficient and powerful reasoners. Continuing research in integrated AI and DA systems (both AI→DA and DA→AI) will provide useful mechanisms as well as a context for applying preference modelers. And ideas from other knowledge-based systems for mathematical modeling should be transferable to the preference modeling task.

A distinguishing architectural feature of programs that employ preference modeling is the separation of reasoning about consequences of actions from reasoning about desirability of outcomes. In itself this is a strong constraint on reasoning style; it implies that decision making programs can never associate beliefs directly with actions. For example, a program for medical therapy cannot implement policies of prescribing drugs based only on observations. Rather, such programs must reason about the implications of these observations combined with the consequences of possible actions (drug recommendations) in terms of the patient state or other relevant attributes. The resulting outcomes are compared with respect to preference to choose the best recommendation. In practice, we might wish to circumvent most of this reasoning for "obvious" action indicators. A *policy* may be thought of as a conscious decision not to explicitly consider consequences of actions and their desirability in certain common situations. This notion will have to be developed further before preference modeling can be practical for routine decision making.

In the introductory chapter I noted that research on URP could be viewed as a step toward the development of a "calculus" for preference assertions. To perform preference modeling, programs will need a language to express their objective structures; URP's technical vocabulary is an early design of that language. The ability to define the language also affords control over the scope of the preference reasoning task. Applications that refer to a constrained set of outcomes or possible preference structures can get by with only a subset of the language and therefore a smaller preference reasoner. This is another way to combat the complexity of the most general preference modeling problem. Thus, some future work may be applied to the identification of simpler preference reasoning tasks sufficient for specific applications.

Once proven preference reasoning capabilities are developed, it would not be surprising to see preferences models used for purposes other than decision making.

The value of explicit preferences for explanation and justification of program behavior has already been discussed. In addition, preference representations may prove a useful basis for heuristics for directing search or information acquisition, or for some kinds of cognitive modeling. Programs that model users' beliefs (for tutoring or interactive consultation, for example) might also benefit from models of users' preferences.

References

[1] Vijay S. Bawa.
Stochastic dominance: A research bibliography.
Management Science 28:698-712, 1982.

[2] J. Robert Beck, Jerome P. Kassirer, and Stephen G. Pauker.
A convenient approximation of life expectancy (the 'DEALE') I. Validation of the method.
American Journal of Medicine 73:883-888, 1982.

[3] David E. Bell.
A utility function for time streams having inter-period dependencies.
Operations Research 25:448-458, 1977.

[4] David E. Bell.
Interpolation Independence.
In S. Zionts (editor), *Lecture notes in economics and mathematical systems.* Volume 155: *Multiple Criteria Problem Solving*, pages 1-7. Springer Verlag, 1978.

[5] David E. Bell.
Multiattribute utility functions: Decompositions using interpolation.
Management Science 25:744-753, 1979.

[6] David E. Bell.
Consistent assessment procedures using conditional utility functions.
Operations Research 27:1054-1066, 1979.

[7] David E. Bell.
Components of risk aversion.
In J. P. Brans (editor), *Operational Research '81*. North-Holland, Amsterdam, 1981.

[8] David E. Bell.
Regret in decision making under uncertainty.
Operations Research 30:961-981, 1982.

[9] David Bell.
Potential contributions to decision analysis.
Decision Sciences 13:534-540, 1982.

[10] David E. Bell.
Risk premiums for decision regret.
Management Science 29:1156-1166, 1983.

[11] David E. Bell.
Disappointment in decision making under uncertainty.
Operations Research 33:1-27, 1985.

[12] James S. Bennett and Robert S. Engelmore.
SACON: A knowledge-based consultant for structural analysis.
In *Sixth International Joint Conference on Artificial Intelligence*, pages 47-49. 1979.

[13] W. W. Bledsoe.
A new method for proving certain Presburger formulas.
In *Fourth International Joint Conference on Artificial Intelligence*, pages 15-21. 1975.

[14] Robert L. Blum.
Lecture Notes in Medical Informatics. Volume 19: *Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project.*
Springer-Verlag, 1982.

[15] Alain Bonnet and Claude Dahan.
Oil-well data interpretation using expert system and pattern recognition technique.
In *Eighth International Joint Conference on Artificial Intelligence*, pages 185-189. 1983.

[16] Hank Bromley.
A program for therapy of acid-base and electrolyte disorders.
Technical Memo 240, MIT Laboratory for Computer Science, 1983.

[17] Rodney A. Brooks.
Symbolic reasoning among 3-d models and 2-d images.
Artificial Intelligence 17:285-348, 1981.

[18] Eugene Charniak and Drew McDermott.
Introduction to Artificial Intelligence.
Addison-Wesley, Reading, MA, 1985.

[19] William J. Clancey.
The epistemology of a rule-based expert system—A framework for explanation.
Artificial Intelligence 20:215-251, 1983.

[20] Paul R. Cohen and Mark D. Lieberman.
A report on FOLIO: An expert assistant for portfolio managers.
In *Eighth International Joint Conference on Artificial Intelligence*, pages 212-214. 1983.

[21] J. Cuena.
The use of simulation models and human advice to build an expert system for the defense and control of river floods.
In *Eighth International Joint Conference on Artificial Intelligence*, pages 246-249. 1983.

[22] Division of Clinical Decision Making, Tufts-New England Medical Center.
Clinical Decision Consults.
1978-1985.
Large file cabinet.

[23] Jon Doyle.
A model for deliberation, action, and introspection.
Technical Report 581, MIT Artificial Intelligence Laboratory, 1980.

[24] Van H. Dunn.
Grand Rounds, Beth Israel Hospital.
1984.
Unpublished decision analysis consult report, Division of Clinical Decision Making, Tufts-New England Medical Center.

[25] Jehoshua Eliashberg and John R. Hauser.
A measurement error approach for modeling consumer risk preference.
Management Science 31:1-25, 1985.

[26] Peter H. Farquhar.
A fractional hypercube decomposition theorem for multiattribute utility functions.
Operations Research 23:941-967, 1975.

[27] Peter H. Farquhar.
A survey of multiattribute utility theory and applications.
In Martin K. Starr and Milan Zeleny (editors), *TIMS Studies in the Management Sciences.* Volume 6: *Multiple Criteria Decision Making*, pages 59-89. North-Holland, 1977.

[28] Peter H. Farquhar.
Advances in multiattribute utility theory.
Theory and Decision 12:381-394, 1980.

[29] Peter H. Farquhar.
Multivalent preference structures.
Mathematical Social Sciences 1:397-408, 1981.

[30] Peter H. Farquhar.
Research directions in multiattribute utility analysis.
In Pierre Hansen (editor), *Lecture notes in economics and mathematical systems.* Volume 209: *Essays and Surveys on Multiple Criteria Decision Making*, pages 63-85. Springer-Verlag, 1983.

[31] Jerome A. Feldman and Robert F. Sproull.
Decision theory and artificial intelligence II: The hungry monkey.
Technical Report 2, University of Rochester Department of Computer Science, 1975.

[32] Jerome A. Feldman and Yoram Yakimovsky.
Decision theory and artificial intelligence: I. A semantics-based region analyzer.
Artificial Intelligence 5:349-371, 1974.

[33] Baruch Fischhoff.
Predicting Frames.
Journal of Experimental Psychology: Learning, Memory, and Cognition 9:103-116, 1983.

[34] Peter C. Fishburn.
von Neumann-Morganstern utility functions on two attributes.
Operations Research 22:35-45, 1974.

[35] Peter C. Fishburn.
The Foundations of Expected Utility.
D. Reidel Publishing Co., Dordrecht, Holland, 1982.

[36] Peter C. Fishburn and Ralph L. Keeney.
Seven independence concepts and continuous multiattribute utility functions.
Journal of Mathematical Psychology 11:294-327, 1974.

[37] Peter C. Fishburn and Ralph L. Keeney.
Generalized utility independence and some implications.
Operations Research 23:928-940, 1975.

[38] Peter C. Fishburn and Raymond G. Vickson.
Theoretical foundations of stochastic dominance.
In G. A. Whitmore and M. C. Findlay (editors), *Stochastic Dominance: An Approach to Decision Making Under Risk.* D. C. Heath and Company, Lexington, MA, 1978.

[39] Kenneth Dale Forbus.
Qualitative process theory.
Technical Report 789, MIT Artificial Intelligence Laboratory, 1984.

[40] Dennis G. Fryback and Ralph L. Keeney.
Constructing a complex judgmental model: An index of trauma severity.
Management Science 29:869-883, 1983.

[41] Amiram Gafni and George W. Torrance.
Risk attitude and time preference in health.
Management Science 30:440-451, 1984.

[42] William A. Gale and Daryl Pregibon.
An expert system for regression analysis.
In *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface,* pages 110-117. 1982.

[43] William A. Gale and Daryl Pregibon.
Artificial intelligence research in statistics.
AI Magazine 5(4):72-75, 1984.

[44] David H. Gustafson, Dennis G. Fryback, Jerry H. Rose, et al.
An evaluation of multiple trauma severity indices created by different index development strategies.
Medical Care 21:674-691, 1983.

[45] G. H. Hardy, J. E. Littlewood, and G. Pôlya.
Inequalities.
Cambridge University Press, London, 1952.
Second edition (first was 1934).

[46] John R. Hauser.
Consumer preference axioms: Behavioral postulates for describing and predicting stochastic choice.
Management Science 24:1331-1341, 1978.

[47] John C. Hershey, Howard C. Kunreuther, and Paul J. H. Schoemaker.
Sources of bias in assessment procedures for utility functions.
Management Science 28:936-954, 1982.

[48] John C. Hershey and Paul J. H. Schoemaker.
Probability vs. certainty equivalence methods in utility measurement: Are they equivalent?.
Technical Report, Center for Decision Research, University of Chicago, 1984.
To appear in *Management Science.*

[49] Michael C. Higgins, Harold C. Sox, Jr., Keith I. Marton, et al.
Clinical utility analysis of patient preferences for prostate cancer treatment outcomes.
1982.
Unpublished project report.

[50] J. P. Hollenberg.
The decision tree builder: An expert system to simulate medical prognosis and management.
1984.
Abstract to appear in *Medical Decision Making.*

[51] Samuel Holtzman.
A decision aid for patients with end-stage renal disease.
Technical Report, Stanford University Department of Engineering-Economic Systems, 1983.

[52] Samuel Holtzman.
Intelligent Decision Systems.
PhD thesis, Stanford University, March, 1985.

[53] Ronald A. Howard.
On fates comparable to death.
Management Science 30:407-422, 1984.

[54] Ronald A. Howard and James E. Matheson.
Influence diagrams.
In Ronald A. Howard and James E. Matheson (editors), *The Principles and Applications of Decision Analysis.* SDG Publications, Menlo Park, CA, 1984.

[55] Daniel Kahneman and Amos Tversky.
Prospect theory: An analysis of decision under risk.
Econometrica 47:263-291, 1979.

[56] Uday S. Karmarkar.
Subjectively weighted utility: A descriptive extension of the expected utility model.
Organizational Behavior and Human Performance 21:61-72, 1978.

[57] Jerome P. Kassirer.
Adding insult to injury: Usurping patients' prerogatives.
New England Journal of Medicine 308:898-901, 1983.

[58] Ralph L. Keeney.
Risk independence and multiattributed utility functions.
Econometrica 41:27-34, 1973.

[59] Ralph L. Keeney.
Measurement scales for quantifying attributes.
Behavioral Science 26:29-36, 1981.

[60] Ralph L. Keeney.
Analysis of preference dependencies among objectives.
Operations Research 29:1105-1120, 1981.

[61] Ralph L. Keeney and Howard Raiffa.
Decisions with Multiple Objectives: Preferences and Value Tradeoffs.
John Wiley and Sons, 1976.

[62] Ralph L. Keeney and Alan Sicherman.
Assessing and analyzing preferences concerning multiple objectives: An interactive computer program.
Behavioral Science 21:173-182, 1976.

[63] Craig W. Kirkwood.
Parametrically dependent preferences for multiattributed consequences.
Operations Research 24:92-103, 1976.

[64] Craig W. Kirkwood and Rakesh K. Sarin.
Ranking with partial information: A method and an application.
Operations Research 33:38-48, 1985.

[65] Donald W. Kosy and Ben P. Wise.
Self-explanatory financial planning models.
In *National Conference on Artificial Intelligence*, pages 176-181. 1984.

[66] David H. Krantz, R. Duncan Luce, Patrick Suppes, et al.
Foundations of Measurement.
Academic Press, New York, 1971.

[67] Benjamin Kuipers.
Commonsense reasoning about causality: Deriving behavior from structure.
Artificial Intelligence 24:169-204, 1984.

[68] John C. Kunz.
Analysis of physiological behavior using a causal model based on first principles.
In *National Conference on Artificial Intelligence*, pages 225-228. 1983.

[69] Paul E. Lehner and Michael L. Donnell.
Building decision aids: Exploiting the synergy between decision analysis and artificial intelligence.
Technical Report, PAR Technology Corporation, 1984.

[70] John D. C. Little.
Models and managers: The concept of a decision calculus.
Management Science 16:B466-B485, 1970.

[71] Graham Loomes and Robert Sugden.
Regret theory: An alternative theory of rational choice under uncertainty.
The Economic Journal 92:805-824, 1982.

[72] Mark J. Machina.
The economic theory of individual behavior toward risk: Theory, evidence and new directions.
Technical Report 433, Institute for Mathematical Studies in the Social Sciences, Stanford University, 1983.

[73] David Allen McAllester.
Reasoning Utility Package user's manual.
AI Memo 667, MIT Artificial Intelligence Laboratory, 1982.

[74] Drew McDermott.
Data dependencies on inequalities.
In *Eighth International Joint Conference on Artificial Intelligence*, pages 266-269. 1983.

[75] Barbara J. McNeil and Stephen G. Pauker.
The patient's role in assessing the value of diagnostic tests.
Radiology 132:605-610, 1979.

[76] Barbara J. McNeil, Stephen G. Pauker, Harold C. Sox, Jr., et al.
On the elicitation of preferences for alternate therapies.
New England Journal of Medicine 306:1259-1262, 1982.

[77] Barbara J. McNeil, Ralph Weichselbaum, and Stephen G. Pauker.
Speech and survival: Tradeoffs between quality and quantity of life in laryngeal cancer.
New England Journal of Medicine 305:982-987, 1981.

[78] Richard F. Meyer.
Preferences over time.
In Ralph L. Keeney and Howard Raiffa (editors), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, chapter 9. John Wiley and Sons, 1976.

[79] K. C. Mosler.
Stochastic dominance decision rules when the attributes are utility independent.
Management Science 30:1311-1322, 1984.

[80] Mancur Olson and Martin J. Bailey.
Positive time preference.
Journal of Political Economy 89:1-25, 1981.

[81] George R. Parkerson, Jr., Stephen H. Gehlbach, and Edward H. Wagner.
The Duke-UNC health profile: An adult health status instrument for primary care.
Medical Care 19:806-828, 1981.

[82] Stephen G. Pauker and Jerome P. Kassirer.
Clinical decision analysis by personal computer.
Archives of Internal Medicine 141:1831-1837, 1981.

[83] Stephen G. Pauker, Susan P. Pauker, and Barbara J. McNeil.
Implications of parents' attitudes on alternative policies for prenatal diagnosis.
In Barbara J. McNeil and Ernest G. Cravalho (editors), *Critical Issues in Medical Technology.* Auburn House Publishing Company, Boston, 1982.

[84] John W. Payne, Dan J. Laughhunn, and Roy Crum.
Multiattribute risky choice behavior: The editing of complex prospects.
Management Science 30:1350-1361, 1984.

[85] Joseph S. Pliskin, Donald S. Shepard, and Milton C. Weinstein.
Utility functions for life years and health status.
Operations Research 28:206-224, 1980.

[86] Howard Raiffa, William B. Schwartz, and Milton C. Weinstein.
Evaluating health effects of societal decisions and programs.
Discussion paper 48D, John Fitzgerald Kennedy School of Government, 1977.

[87] Jean-Louis Roos.
Artificial intelligence and macro-economy, an application.
In *Eighth International Joint Conference on Artificial Intelligence*, pages 250-251. 1983.

[88] Cynthia J. Rutherford, Byron Davies, Arnold I. Barnett, et al.
A computer system for decision analysis in Hodgkins Disease.
Technical Report 271, MIT Laboratory for Computer Science, 1981.

[89] Elisha Sacks.
Qualitative mathematical reasoning.
Technical Report 329, MIT Laboratory for Computer Science, 1985.

[90] Elisha Sacks.
Qualitative mathematical reasoning.
In *Ninth International Joint Conference on Artificial Intelligence.* 1985.

[91] Rakesh Kumar Sarin.
Interactive Procedures for Evaluation of Multi-Attributed Alternatives.
PhD thesis, University of California, Los Angeles, 1975.

[92] Leonard J. Savage.
The Foundations of Statistics.
Dover Publications, New York, 1972.
Second revised edition (first was 1954).

[93] Paul J. H. Schoemaker.
The expected utility model: Its variants, purposes, evidence and limitations.
Journal of Economic Behavior 20:529-563, 1982.

[94] Paul J. H. Schoemaker and C. Carter Waid.
An experimental comparison of different approaches to determining weights in additive utility models.
Management Science 28:182-196, 1982.

[95] Edward Hance Shortliffe.
Computer-Based Medical Consultations: MYCIN.
American Elsevier, New York, 1976.

[96] Robert E. Shostak.
On the SUP-INF method for proving Presburger formulas.
Journal of the ACM 24:529-543, 1977.

[97] Martin K. Starr and Milan Zeleny.
MCDM—State and future of the arts.
In Martin K. Starr and Milan Zeleny (editors), *TIMS Studies in the Management Sciences.* Volume 6: *Multiple Criteria Decision Making*, pages 5-29. North-Holland, 1977.

[98] Gerald Jay Sussman and Guy Lewis Steele Jr.
CONSTRAINTS—A language for expressing almost-hierarchical descriptions.
Artificial Intelligence 14:1-39, 1980.

[99] William R. Swartout.
XPLAIN: A system for creating and explaining expert consulting programs.
Artificial Intelligence 21:285-325, 1983.

[100] Peter Szolovits and Stephen G. Pauker.
Categorical and probabilistic reasoning in medical diagnosis.
Artificial Intelligence 11:115-144, 1978.

[101] Peter Szolovits and Stephen G. Pauker.
Computers and clinical decision making: Whether, how, and for whom?
Proceedings of the IEEE 67:1224-1226, 1979.

[102] Richard Thaler.
Toward a positive theory of consumer choice.
Journal of Economic Behavior and Organization 1:39-60, 1980.

[103] Richard H. Thaler and H. M. Shefrin.
An economic theory of self-control.
Journal of Political Economy 89:392-406, 1981.

[104] George W. Torrance, Michael H. Boyle, and Sargent P. Horwood.
Application of multi-attribute utility theory to measure social preferences for health states.
Operations Research 30:1043-1069, 1982.

[105] A. Tversky.
On the elicitation of preferences: Descriptive and prescriptive considerations.
In David E. Bell, Ralph L. Keeney, and Howard Raiffa (editors), *Conflicting Objectives in Decisions.* John Wiley & Sons, 1977.

[106] Amos Tversky and Daniel Kahneman.
The framing of decisions and the psychology of choice.
Science 211:453-458, 1981.

[107] John von Neumann and Oskar Morganstern.
Theory of Games and Economic Behavior.
Princeton University Press, Princeton, 1953.
Third edition (first was 1944).

[108] David Waltz.
Understanding line drawings of scenes with shadows.
In Patrick Henry Winston (editor), *The Psychology of Computer Vision.* McGraw-Hill, 1975.

[109] Milton C. Weinstein and Robert J. Quinn.
Psychological considerations in valuing health risk reductions.
Natural Resources Journal 23:659-673, 1983.

[110] Sholom Weiss, Casimir Kulikowski, Chidanand Apté, et al.
Building expert systems for controlling complex programs.
In *National Conference on Artificial Intelligence*, pages 322-326. 1982.

[111] Michael P. Wellman.
Reasoning about assumptions underlying mathematical models.
1985.
Unpublished manuscript.

[112] G. A. Whitmore and M. C. Findlay (editors).
Stochastic Dominance: An Approach to Decision Making Under Risk.
D. C. Heath and Company, Lexington, MA, 1978.

[113] Robert D. Willig.
Risk invariance and ordinally additive utility functions.
Econometrica 45:621-640, 1977.

[114] Robert L. Winkler.
Research directions in decision making under uncertainty.
Decision Sciences 13:517-533, 1982.

A. Qualitative Behaviors of Unidimensional Utility Functions

Reasoning about qualitative behaviors of single-attribute utility functions was discussed in chapter 3. This appendix describes the qualitative behaviors currently implemented in URP.

A.1 Behavior Descriptions

monotonic: u's derivative is either nonpositive or nonnegative over the entire domain

strict-monotonic: u's derivative is either positive or negative

monotonic-increasing: u's derivative is strictly positive

monotonic-nondecreasing: u's derivative is nonnegative

monotonic-decreasing: u's derivative is strictly negative

monotonic-nonincreasing: u's derivative is nonpositive

risk-averse: u is concave (second derivative is negative)

risk-neutral: u is linear (zero second derivative); the functional form `linear` applies

risk-prone: u is convex (postive second derivative)

constant-risk-averse: the risk aversion function (r or q from 3.1 or 3.2) is constant; the functional form `exponential` applies

constant-proportional-risk-posture: the proportional risk function $\rho(x) = xr(x)$ (or $xq(x)$ for decreasing u) is constant; one of the functional forms `power` or `log` applies (intensity of risk aversion distinguishes the cases)

increasing-risk-posture: the risk function is **monotonic-increasing**

decreasing-risk-posture: the risk function is **monotonic-decreasing**

The mathematical relations between the behaviors and u is encoded in the *pos*, *nec*, and *enf* procedures described in section 3.3.3. Logical relations among the behaviors are encoded in RUP. The functional forms mentioned above appear in table 3.1.

A.2 Terminological Definitions

constant-risk-averse	$\equiv$	**constant-risk-posture** $\wedge$ **risk-averse**
constant-risk-prone	$\equiv$	**constant-risk-posture** $\wedge$ **risk-prone**
constant-proportional-risk-averse	$\equiv$	**constant-proportional-risk-posture** $\wedge$ **risk-averse**
constant-proportional-risk-prone	$\equiv$	**constant-proportional-risk-posture** $\wedge$ **risk-prone**
increasing-risk-averse	$\equiv$	**risk-averse** $\wedge$ **increasing-risk-posture**
increasing-risk-prone	$\equiv$	**risk-prone** $\wedge$ **increasing-risk-posture**
decreasing-risk-averse	$\equiv$	**risk-averse** $\wedge$ **decreasing-risk-posture**
decreasing-risk-prone	$\equiv$	**risk-prone** $\wedge$ **decreasing-risk-posture**

B. Axiom Schemata

Here is a complete alphabetized listing of the independence axioms currently known to URP. Some of these are not yet used in any significant reasoning (they do not appear in any theorems). The representation is described in section 5.1.1, and the meanings of the individual axioms are discussed in chapter 6. The "`:definition`" slots are not filled in carefully (or at all) since there is as yet no explanation facility. Definitions from the literature can be found in the document appearing in the "`:source`" slot; pointers are given in the references section.

```
(defaxiom 'additive-independence
  :source "Keeney & Raiffa p. 295"
  :special-case-of '(mutual-utility-independence)
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'additive-value-function
  :definition "can use additive form for value function"
  :source "Farquhar 77 p. 64"
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'bilateral-form
  :definition "functional form"
  :source "Farquhar 77 p. 71"
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'bilateral-independence
  :source "Farquhar 77 p. 71"
  :special-case-of '(generalized-bilateral-independence
                     joint-interpolation-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))
```

```
(defaxiom 'conditional-preferential-independence
  :definition "PI given Y-(not ij)'"
  :source "Keeney and Raiffa p. 334"
  :arg-count 3
  :arg-relns '((exclusive-exhaustive-nonempty-subsets 1 2 3)))

(defaxiom 'conditional-utility-independence
  :definition "UI given Y-(not ij)'"
  :source "Keeney and Raiffa p. 334"
  :arg-count 3
  :arg-relns '((exclusive-exhaustive-nonempty-subsets 1 2 3)))

(defaxiom 'conditional-value-independence
  :definition "VI given Y-(not ij)'"
  :source "Keeney and Raiffa p. 336"
  :arg-count 3
  :arg-relns '((exclusive-exhaustive-nonempty-subsets 1 2 3)))

(defaxiom 'generalized-bilateral-independence
  :definition "BI with reversals allowed"
  :source "Farquhar 77 p. 71"
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'generalized-preferential-independence
  :source "Fishburn & Keeney 74 p. 299"
  :special-case-of '(indifference-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'generalized-utility-independence
  :source "Farquhar 77 p. 69"
  :special-case-of '(generalized-preferential-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))
```

```
(defaxiom 'indifference-independence
  :source "Fishburn & Keeney 74 p. 299"
  :restrictions "s-c-o GPI requires Axiom 1 from source"
  :special-case-of '(weak-indifference-independence
                     generalized-preferential-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'interpolation-independence
  :definition "see Bell 79b p. 1056"
  :source "Bell 78, 79a, 79b"
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'multilinear-form
  :definition "functional form"
  :source "Keeney & Raiffa p. 293"
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'multilinear-generalization-form
  :definition "functional form"
  :source "Bell 79a p. 748"
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'multiplicative-form
  :definition "functional form"
  :source "Keeney & Raiffa p. 289"
  :special-case-of '(multilinear-form)
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'mutual-preferential-independence
  :definition "every subset PI its complement"
  :source "Keeney and Raiffa p. 111"
  :arg-count 1
  :arg-relns '((nonempty 1)))
```

```
(defaxiom 'mutual-utility-independence
  :source "Keeney & Raiffa p. 289"
  :arg-count 1
  :arg-relns '((nonempty 1)))

(defaxiom 'parametric-independence
  :source "Kirkwood"
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'preferential-independence
  :source "Keeney & Raiffa p. 109"
  :special-case-of '(generalized-preferential-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'utility-independence
  :source "Keeney & Raiffa p. 284"
  :special-case-of '(preferential-independence
                     generalized-utility-independence
                     bilateral-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))

(defaxiom 'value-independence
  :source "Fishburn & Keeney 74 p. 297"
  :restrictions "special-case of UI requires (5) from source"
  :special-case-of '(utility-independence)
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2))
  :op-prop '(symmetric))

(defaxiom 'weak-indifference-independence
  :source "Fishburn & Keeney 74 p. 300"
  :arg-count 2
  :arg-relns '((exclusive-nonempty-subsets 1 2)))
```

```
(deftheorem 'theorem009
  :source "Keeney & Raiffa p. 316"
  :premise '(for-some-set Y1 in (subset X)
              (for-some-set Y2 in (overlapping-subset Y1 X)
                ((utility-independence Y1 (difference X Y1))
                 (utility-independence Y2 (difference X Y2)))))
  :consequent '((utility-independence (union Y1 Y2)
                                      (difference X (union Y1 Y2)))
              (utility-independence (intersection Y1 Y2)
                                    (difference X (intersection Y1 Y2)))
              (utility-independence (sym-difference Y1 Y2)
                                    (difference X
                                                (sym-difference Y1 Y2)))
              (utility-independence (difference Y1 Y2)
                                    (difference X (difference Y1 Y2)))
              (utility-independence (difference Y2 Y1)
                                    (difference X (difference Y2 Y1)))))


(deftheorem 'theorem010
  :source "Keeney & Raiffa p. 112"
  :premise '(for-some-set Y in (subset X)
              (for-some-set Z in (overlapping-subset Y X)
                ((preferential-independence Y (difference X Y))
                 (preferential-independence Z (difference X Z)))))
  :consequent
  '((preferential-independence (union Y Z) (difference X (union Y Z)))
    (preferential-independence (intersection Y Z)
                               (difference X (intersection Y Z)))
    (preferential-independence (sym-difference Y Z)
                               (difference X (sym-difference Y Z)))
    (preferential-independence (difference Y Z)
                               (difference X (difference Y Z)))
    (preferential-independence (difference Z Y)
                               (difference X (difference Z Y)))))


(deftheorem 'theorem012
  :source "Keeney & Raiffa p. 289"
  :premise '(for-some-set Z in (subset X)
                    ((mutual-utility-independence X)))
  :consequent '((utility-independence Z (difference X Z))))
```

```
(deftheorem 'theorem013
  :source "Farquhar 77 Thm. 1"
  :premise '(for-some k in X
              (for-all j in (difference X (set-of k))
                       ((preferential-independence
                          (set-of j k) (difference X (set-of j K))))))
  :consequent '((additive-value-function X)))


(deftheorem 'theorem014
  :source "Farquhar 77 Thm. 5"
  :premise '(for-all i in X
              ((generalized-bilateral-independence
                             (set-of i) (difference X (set-of i)))))
  :consequent '((bilateral-form X)))


(deftheorem 'theorem016
  :source "Bell 79a p. 748"
  :premise '(for-all i in X
              ((joint interpolation-independence
                             (set-of i) (difference X (set-of i)))))
  :consequent '((multilinear-generalization-form X)))


(deftheorem 'theorem017
  :source "Fishburn and Keeney 74 Thm 1."
  :restrictions "Axiom 1, Y1 essential"
  :premise '(for-some-set Y1
              (for-some-set Y2
                 (for-some-set Y3
                    ((weak-indifference-independence (union Y1 Y2)
                                                      Y3)
                     (value-independence Y1 (union Y2 Y3))))))
  :consequent '((value-independence (union Y1 Y2) Y3)
              (value-independence (union Y1 Y3) Y2)))
```

```
(deftheorem 'theorem018
  :source "Fishburn and Keeney 74 Thm 2."
  :restrictions "Axiom 1, Y1 essential"
  :premise '(for-some-set Y1
               (for-some-set Y2
                  (for-some-set Y3
                     ((weak-indifference-independence (union Y1 Y2)
                                                      Y3)
                      (utility-independence Y1 (union Y2 Y3))))))
  :consequent '((utility-independence (union Y1 Y2) Y3)))


(deftheorem 'theorem019
  :source "Fishburn and Keeney 74 Thm. 3"
  :restrictions "Axiom 1, Y1 + Y2 essential, see also page 303"
  :premise '(for-some-set Y1
               (for-some-set Y2
                  (for-some-set Y3
                     ((weak-indifference-independence (union Y1 Y2)
                                                      Y3)
                      (utility-independence (union Y1 Y3) Y2)))))
  :consequent '((utility-independence Y3 (union Y1 Y2))
               (generalized-utility-independence (union Y1 Y2) Y3)))


(deftheorem 'theorem020
  :source "Fishburn and Keeney 74 Thm. 3 corollary (at bottom)"
  :restrictions "Axiom 1, Y1 + Y2 essential, see also page 303"
  :premise '(for-some-set Y1
               (for-some-set Y2
                  (for-some-set Y3
                     ((preferential-independence (union Y1 Y2) Y3)
                      (utility-independence (union Y1 Y3) Y2)))))
  :consequent '((utility-independence (union Y1 Y2) Y3)))
```

```
(deftheorem 'theorem021
  :source "Fishburn and Keeney 74 corollary p. 308"
  :restrictions "Axiom 1, Y1 essential"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      ((generalized-utility-independence Y1
                                                          (union Y2 Y3))
                       (preferential-independence (union Y1 Y2) Y3)
                       (preferential-independence (union Y1 Y3) Y2)))))
  :consequent '((utility-independence Y1 (union Y2 Y3))
              (utility-independence (union Y1 Y2) Y3)
              (utility-independence (union Y1 Y3) Y2)))


(deftheorem 'theorem023
  :source "Keeney and Raiffa Thm. 6.15"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      ((utility-independence Y1 (union Y2 Y3))))))
  :consequent '((utility-independence Y1 Y2)))


(deftheorem 'theorem024
  :source "Keeney and Raiffa Thm. 6.17"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      ((utility-independence Y1 Y2)
                       (conditional-utility-independence Y1 Y3 Y2)))))
  :consequent '((utility-independence Y1 (union Y2 Y3))))
```

```
(deftheorem 'theorem025
  :source "Keeney and Raiffa Thm. 6.18"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      ((utility-independence (union Y1 Y2) Y3)
                       (conditional-preferential-independence
                                                            Y1 Y2 Y3)))))
  :consequent '((preferential-independence Y1 (union Y2 Y3))))


(deftheorem 'theorem026
  :source "Keeney and Raiffa Thm. 6.19"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      ((utility-independence (union Y1 Y2) Y3)
                       (conditional-utility-independence Y1 Y2 Y3)))))
  :consequent '((utility-independence Y1 (union Y2 Y3))))


(deftheorem 'theorem027
  :source "special case of conditional UI"
  :premise '(for-some-set Y1 in X
              (for-some-set Y2 in (difference X Y1)
                  ((utility-independence Y1 Y2))))
  :consequent '((conditional-utility-independence
                              Y1 Y2 (difference X (union Y1 Y2)))))


(deftheorem 'theorem028
  :source "special case of conditional PI"
  :premise '(for-some-set Y1 in X
              (for-some-set Y2 in (difference X Y1)
                  ((preferential-independence Y1 Y2))))
  :consequent '((conditional-preferential-independence
                              Y1 Y2 (difference X (union Y1 Y2)))))
```

```
(deftheorem 'theorem029
  :source "Bell 77 Lemma 2"
  :restriction "I dropped the X4 but I think this still works"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3 in (subset (complement Y2))
                      ((joint utility-independence Y1 (union Y2 Y3))))))
  :consequent '((joint utility-independence Y1 Y2)
              (joint utility-independence Y1 Y3)))


(deftheorem 'theorem030
  :source "Bell 77 Lemma 2 extension"
  :restriction "I dropped the X4 but I think this still works"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3 in (subset (complement Y2))
                      ((utility-independence Y1 Y2)
                       (utility-independence Y1 Y3)))))
  :consequent '((utility-independence Y1 (union Y2 Y3))))


(deftheorem 'theorem031
  :source "Fishburn & Keeney 75 Lemma 1 p. 931"
  :restriction "Y1 essential"
  :premise '(for-some-set Y1
              (for-some-set Y2
                  (for-some-set Y3
                      (for-some-set Y4
                          ((generalized-utility-independence
                                    (union Y1 Y2) (union Y3 Y4))
                           (generalized-utility-independence
                                    (union Y1 Y3) (union Y2 Y4)))))))
  :consequent '((generalized-utility-independence
                      (union Y1 (union Y2 Y3)) Y4)))
```

```
(deftheorem 'theorem032
  :source "Fishburn & Keeney 75 Thm 1 p. 934"
  :restriction "Yi essential"
  :premise '(for-some i in X
              (for-all j in (difference X (set-of i))
                ((generalized-utility-independence
                     (set-of i j) (difference X (set-of i j))))))
  :consequent '((multiplicative-form X)))


(deftheorem 'theorem033
  :source "Keeney 74 Lemma 2 p. 24"
  :premise '(for-some-set Y1
              (for-some-set Y2
                (for-some-set Y3
                  (for-some-set Y4
                    ((utility-independence (union Y1 Y2)
                                           (union Y3 Y4))
                     (utility-independence (union Y1 Y3)
                                           (union Y2 Y4)))))))
  :consequent '((utility-independence (union Y1 (union Y2 Y3)) Y4)))
```

D. Set Membership Constraints

The following is a complete list of set membership constraints appearing in the current set of URP multiattribute utility theorems. Each is defined in terms of the primitive constraints **union** and **complement**, described in section 5.3.2.

The expressions defined below are shown as they appear in URP theorems. The set denoted by the expression is called Z in the definition. When notation is given it is for use in subsequent constraint definitions. The predicate *nonempty* on a set is used to indicate that **min-size** for that set is restricted to be greater than or equal to one.[1]

(**intersection** `Y1 Y2`)

notation: $Z = Y_1 \cap Y_2$

definition: $Z = \overline{\overline{Y_1} \cup \overline{Y_2}}$

(**difference** `X Y`)

notation: $Z = X - Y$

definition: $Z = \overline{\overline{X} \cup Y}$

(**sym-difference** `X Y`)

definition: $Z = (X - Y) \cup (Y - X)$

(**subset** `X`)

notation: $Z \subseteq X$

definition: $X = X \cup Z$

(**nonempty-subset** `X`)

definition: $Z \subseteq X$; *nonempty*(Z)

(**proper-subset** `X`)

definition: $Z \subseteq X$; *nonempty*$(X - Z)$

(**overlapping-subset** `Y X`)

[1]Strictly speaking, *nonempty* is not a constraint since it does not appear as an object in the network. It only influences the initial value for **min-size** during network construction.

definition: $Z \subseteq X$; $Y \subseteq X$; $nonempty(Z - Y)$;
$nonempty(Y - Z)$; $nonempty(Y \cap Z)$;
$nonempty(X - (Y \cup Z))$

The following two constraints are used exclusively for type-checking the arguments to URP axioms. They appear in the **arg-relns** slot of the axiom schemata.

(exclusive-nonempty-subsets Y1 Y2**)**

definition: $Y_1 \subseteq \overline{Y_2}$; $nonempty(Y_1)$; $nonempty(Y_2)$

(exclusive-exhaustive-nonempty-subsets Y1 Y2 Y3**)**

definition: $Y_1 \subseteq \overline{Y_2}$; $Y_3 = \overline{Y_1 \cup Y_2}$; $nonempty(Y_1)$;
$nonempty(Y_2)$; $nonempty(Y_3)$

URP can also handle constraints that relate set nodes to element nodes. The only one used explicitly in theorems is **set-of** (although the definition below uses two arguments, the constraint may be used for an arbitrary number of element variables):

(set-of i j**)**

definition: $Z = \{i, j\}$

E. Multiattribute Functional Forms

This appendix lists the multiattribute functional forms that may be part of decompositions generated by URP. Each form (named for the independence axiom it is associated with) is written in a mathematical notation and a plausible URP specification. The specification is for illustration only; no facilities for mathematical reasoning about multiattribute functions has been implemented.

E.1 Two-attribute Forms

Several of URP's binary independence relations are associated directly with two-attribute decompositions. This section lists those conditions, along with a specification for the functional form they imply. This listing is almost identical to table 1 from Bell [6] (except for the URP specification, of course). Conditialization is omitted for unidimensional functions if the conditioning level of the fixed attribute is not significant.

- value-independence

$$au(x) + (1 - a)u(y)$$

```
((+ (* a (u x))
    (* (- 1 a) (u y)))
 (where a (evaluation u (best x) (worst y))))
```

- joint utility-independence

$$au(x) + bu(y) + (1 - a - b)u(x)u(y)$$

```
((+ (* a (u x))
    (* b (u y))
    (* (- 1 a b) (u x) (u y)))
 (where a (evaluation u (best x) (worst y))
        b (evaluation u (worst x) (best y))))
```

- utility-independence

$$au(x) + bu(x_*, y) - bu(x)u(x_*, y) + (1 - a)u(x)u(x^*, y)$$

```
((+ (* a (u x))
    (* b (u y (worst x)))
    (* (- b) (u x) (u y (worst x)))
    (* (- 1 a) (u x) (u y (best x))))
 (where a (evaluation u (best x) (worst y))
        b (evaluation u (worst x) (best y))))
```

- bilateral-independence

$$au(x, y_*) + bu(y, x_*)$$
$$+\frac{1}{1-a-b} \cdot [(1-b)u(x, y^*) - au(x, y_*)] \cdot [(1-a)u(x, y^*) - bu(y, x_*)]$$

```
((+ (* a (u x))
    (* b (u y (worst x)))
    (* (/ 1 (- 1 a b))
       (- (* (- 1 b) (u x (best y)))
          (* a (u x (worst y))))
       (- (* (- 1 a) (u y (best x)))
          (* b (u y (worst x))))))
 (where a (evaluation u (best x) (worst y))
        b (evaluation u (worst x) (best y))))
```

- joint interpolation-independence

$$au(x, y_*) + bu(y, x_*) - ku(x, y_*)u(y, x_*)$$
$$+(k-a)u(x, y_*)u(y, x^*) + (k-b)u(x, y^*)u(y, x_*) + (1-k)u(x, y^*)u(y, x^*)$$

```
((+ (* a (u x))
    (* b (u y (worst x)))
    (* (- k) (u x (worst y)) (u y (worst x)))
    (* (- k a) (u x (worst y)) (u y (best x)))
    (* (- k b) (u x (best y)) (u y (worst x)))
    (* (- 1 k) (u x (best y)) (u y (best x)))
 (where a (evaluation u (best x) (worst y))
        b (evaluation u (worst x) (best y))))
```

E.2 n-attribute Forms

- Additive:

$$\sum_{i=1}^{n} k_i u_i(x), \quad \text{where} \quad k_i = u(x_i^*, x_{\bar{i}*})$$

```
((sum i (from-to 1 n)
      (* (k i) (u i)))
 (where (k i) (evaluation u (best i) (worst (comp i)))))
```

For the following forms I will use the notation $\mathbf{i}_r$ to stand for a collection of r subscripts, and the symbol S_r to denote the space of all such subscript collections.

- Multiplicative:

$$\sum_{r=1}^{n} \sum_{\mathbf{i}_r \in S_r} k^{r-1} \prod_{i \in \mathbf{i}_r} k_i u_i(x_i)$$

$$\text{where } 1 + k = \textstyle\prod_{i=1}^{n}(1 + k k_i),\ k_i \text{ as above}$$

```
((sum r (from-to 1 n)
      (sum i-subs (subscript-collns-of-size r n)
           (* (expt k (- r 1))
              (product i (subscripts-in i-subs)
                       (* (k i) (u i))))))
 (where (k i) (evaluation u (best i) (worst (comp i)))
        k (iterative-soln (- (product i (from-to 1 n)
                                      (+ 1 (* k (k i))))
                             1))))
```

- Multilinear:

$$\sum_{r=1}^{n} \sum_{\mathbf{i}_r \in S_r} k_{\mathbf{i}_r} \prod_{i \in \mathbf{i}_r} u_i(x_i)$$

$$\text{where } k_{\mathbf{i}_r} = u(x_{\mathbf{i}_r}^*, x_{\bar{\mathbf{i}}_r *}) - \textstyle\sum_{r_0=1}^{r-1} \sum_{\mathbf{i}_{r_0} \in I_{r_0}} k_{\mathbf{i}_{r_0}}$$

```
((sum r (from-to 1 n)
     (sum i-subs (subscript-collns-of-size r n)
          (* (k i-subs)
             (product i (subscripts-in i-subs)
                        (u i)))))
 (where (k i-subs)
        (- (evaluation u (best i-subs) (worst (comp i-subs)))
           (sum r-zero (from-to 1 (- (length i-subs) 1))
                (sum i-subs-zero
                     (subscript-collns-of-size r-zero n)
                     (k i-subs-zero)))))))
```

- Multilinear Generalization:

$$\theta + \sum_{r=1}^{n} \sum_{\mathbf{i}_r \in S_r} k'_{\mathbf{i}_r} \prod_{i \in \mathbf{i}_r} v_i^*(x_i) \prod_{i \notin \mathbf{i}_r} v_{i*}(x_i)$$

where $k'_{\mathbf{i}_r} = u(x^*_{\mathbf{i}_r}, x_{\bar{\mathbf{i}}_r *}) - \theta$, θ is an interpolation constant, v^* and v_* are as defined by Bell [5], page 747.

```
((+ theta
    (sum r (from-to 1 n)
         (sum i-subs (subscript-collns-of-size r n)
              (* (k-prime i-subs)
                 (product i (subscripts-in i-subs)
                            (v-best i))
                 (product i (comp (subscripts-in i-subs))
                            (v-worst i))))))
 (where (k-prime i-subs)
        (- (evaluation u (best i-subs) (worst (comp i-subs)))
           theta)))
```

- Bilateral:

$$\theta + \sum_{r=1}^{n} \sum_{\mathbf{i}_r \in S_r} k'_{\mathbf{i}_r} \prod_{i \in \mathbf{i}_r} v_i^*(x_i) \prod_{i \notin \mathbf{i}_r} v_{i*}(x_i) + \sum_{i=1}^{n} \frac{a_i(1-b_i)\left[u(x_i, x_{\bar{i}*}) - u(x_i, x^*_{\bar{i}})\right]}{1 - a_i - b_i}$$

where a_i and b_i are also defined by Bell [5]

The bilateral form could be specified in a fashion similar to the above.

F. The Health Preference Knowledge Base

This appendix summarizes the health preference knowledge base developed in chapter 8 (plus a few concepts not included there). The representations are built from URP's technical vocabulary, with some special-purpose mechanisms that will be necessary to manage the knowledge base. The concepts encoded here are just for illustration; there is no claim that they are complete or that they represent anyone's actual preferences. This knowledge base has not been implemented for URP.

F.1 Built-in Attributes

```
(defattribute LIFE-YEARS
   :value-set '(real-range 0 max-life-years)
   :scale-type 'ratio
   :unit 'years
   :description
      '(life-years (# of years from NOW until DEATH)
        max-life-years (Largest possible value for life-years)))
```

```
(defattribute PHYSICAL-FUNCTION
   :source Torrance, Boyle, and Horwood
   :value-set '(integer-range 1 6)
   :scale-type 'ordinal
   :unit '((1 (Needs help to get around; no control of limbs))
           (2 (Needs help; needs mechanical aids to walk))
           (3 (Needs help; limitations in physical activity))
           (4 (Can get around; needs mechanical aids))
           (5 (Can get around; limitations in activity))
           (6 (Can get around; no limitations)))
   :description
      '(physical-function (Mobility and physical activity)))
```

(component *physical-function health-status*)

(component *role-function health-status*)

(component *social-emotional-function health-status*)

(component *health-problem health-status*)

Note: These components of health status are taken from the health state classification system of Torrance, Boyle, and Horwood [104].

F.2 Predefined Qualitative Behaviors

(**monotonic-increasing** *life-years*)[1]

(**monotonic-increasing** *physical-function*)

(**monotonic-decreasing** *hospital-confinement*)

(**monotonic-decreasing** *cost*)

(influence+ **decreasing-marginal-value** *life-years* **risk-averse**)

(influence+ **increasing-marginal-value** *life-years* **risk-prone**)

(influence+ **time-preference** *life-years* **risk-averse**)

(influence+ **youth-preference** *life-years* **risk-averse**)

(influence+ **age-preference** *life-years* **risk-prone**)

Note: a positive influence on **risk-averse** is equivalent to a negative influence on **risk-prone**, and vice-versa.

F.3 Predefined Independence Conditions

(**preferential-independence** {*health-status*} {*life-years*})

(**generalized-preferential-independence** {*life-years*} {*health-status*})

Other independence conditions may be inferred from domain concepts using the rules of section F.5.

[1] Following the development in section 8.5, it might be desirable to condition this on a health state other than "brain-dead."

F.4 Special Functional Forms

(**monotonic-transform** *life-years deale-life-expectancy*)

```
(iff (and (joint utility-independence {life-years} {health-status})
          (constant-proportional-tradeoff life-years health-status)
          (risk-neutral life-years))
     (QALY-form-1))

(iff (and (joint utility-independence {life-years} {health-status})
          (constant-proportional-tradeoff life-years health-status))
     (QALY-form-2))

(iff (successive-pairwise-preferential-independence T)
     (from-to i (1 (- n 1))
          (preferential-independence
                    (set-of (nth i T) (nth (+ i 1) T))
                    (difference T (set-of (nth i T) (nth (+ i 1) T))))))

(iff (and (constant-proportional-tradeoff consumption life-years)
          (constant-risk-averse life-years))
     (micromort-model))
```

Notes: The **monotonic-transform** assertion on two attributes states that behaviors that are invariant over monotonic transformations which hold for the first attribute also hold for the second. **QALY-form-1** and **QALY-form-2** correspond to equations 8.2 and 8.3, respectively. The micromort model is Howard's measure for the small-risk value of life [53], mentioned in section 8.6.4.

F.5 Relations to Domain Concepts

The validity of these URP assertions will often depend on the characteristics of the medical situation. Facts of the particular case will indicate which attributes are relevant, features of preference for these attributes, and valid independence assumptions. This section illustrates some simple rule encodings for these relations. Through a mechanism such as this, any of the above assertions may be tied to triggering or exceptional situations.

RULE001:

IF There are no states worse than death
THEN (**joint preferential-independence** {*life-years*} {*health-status*})
(**monotonic-increasing** *life-years*)

RULE002:

IF Financial costs are not borne by the patient
THEN (**value-independence** {*cost*} *everything else*)
(**monotonic-decreasing** *cost*)

RULE003:

IF Patient gets weary of staying in hospital
THEN (**risk-averse** *hospital-confinement*)

RULE004:

IF Patient gets accustomed to staying in hospital
THEN (**risk-prone** *hospital-confinement*)