Chapter 3

Limited Rationality

In the preceding chapters we have seen a number of game playing programs. Having adopted the partial state-space search paradigm, the programs answer the question of what part of the space to search in various ways, from a "flat" depth-*d* search to singular extensions, conspiracy search, and finally the Russell-Wefald and Baum-Smith decision theoretic search techniques. All these techniques are attempting to reconcile the need to make a good decision with the need to search quickly, and this reconciliation requires making the meta-level decision of what to search. But we saw that it was important to be careful not to spend too much time making this meta-level decision, either. So we have a rather complicated set of variables to trade off against each other: quality of move choice, versus time spent searching, versus time spent deciding where to search. It would be desirable to have some kind of theoretical understanding of how to make this trade-off, while keeping in mind that we, as designers, do not want to spend arbitrary amounts of time optimizing our program designs, either. The need for efficiency is really present at all levels of the process of designing, building, and running a game-playing program. We would like to have some kind of higher-level understanding of this efficiency constraint and how exactly it should affect our methodologies, at as many levels as possible.

But before we set forth investigating what would be involved in such an endeavor, it is worthwhile to step back from our domain of games, and consider whether this issue is also important outside that realm—for we should always keep in mind that computer game playing is not our ultimate goal, but is merely a way to focus our attempts to develop better general AI techniques.

Indeed, if we look at the larger context of computer problem solving, planning, and so forth, we find that a similar problem appears throughout, and constitutes a fundamental question in building AI programs, namely, the question of Limited Rationality (LR): How do we efficiently utilize limited reasoning resources, when figuring out the answer to this question also consumes those same resources? This seems at first to be a circular question. And yet, it has been shown (for a particular formalization of this question, in [Lipman-91]) that answers do exist, at least in principle. But the question remains of how to get at them in practice.

This chapter will consider these issues in detail. We will look at some of the related investigations in probability, decision theory and economics. We will also briefly survey the work on limited rationality theory that grew out of Russell and Wefald's experiences with their decision-theoretic search control techniques in game playing. Their work provides many interesting insights, but is not at all the final word on this issue. We will also look at several efforts in other areas of AI to partially formalize the trade-offs of limited rationality. These overviews of existing work will provide the background for our own investigations into the subject.

In my attempt to better understand some of this background, I investigated the issue of whether some of the CGP search control techniques could be applied in another domain, that of clinical decision-making. Instead of large game trees, CDM involves large decision-analysis trees. We will discuss these preliminary investigations in §3.3.

Unfortunately, the outcome of these investigations was the realization that there is a fundamental flaw with the limited rationality techniques developed for CGP, namely that they are (perhaps unsurprisingly) anchored to the same concrete state space paradigm that pervades most of CGP research. When I attempted to generalize the CGP search-control techniques to include a decision-analysis problem, I discovered that the state-space paradigm was not appropriate for that problem, or for reasoning in general. This led to my investigations into a new paradigm of *abstract state-space search*, reported in chapter 4, which attempts to preserve some of the desirable features of the regular state space search while avoiding some of its limitations.

Finally, in the last part of this chapter I present the beginnings of some new theoretical concepts for understanding the problem of limited rationality that are free from the limitations of the state-space paradigm, unlike the formulations used in Russell's group. Whether these concepts can be extended to a complete theory of limited rationality remains to be seen, but at least, they provide an interesting place at which to begin.

Now, let us start by looking at some of the history behind the issue of limited rationality.

3.1 History of LR in Decision Theory and Economics

Economists have long been concerned with the problem of modeling the behavior of agents, groups and societies. Their investigations have often been conducted by modeling the reasoning process performed by these entities in order to choose their actions. In order to keep things simple, these models of rationality often made strong assumptions about the deductive powers of the entities. Such models were found to be unrealistic in ways that damaged the quality of the economic predictions derived from the models.

Thus, Simon introduced the notion of "bounded rationality," the study of models of rationality in which entities may be limited by their internal computational capabilities as well as by the structure of their environment:

> ...some of the constraints that must be taken as givens in an optimization problem may be physiological and psychological limitations of the organism (biologically defined) itself. For example, the maximum speed at which an organism can move establishes a boundary on the set of its available behavior alternatives. Similarly, limits on the computational capacity may be important constraints entering into the definition of rational choice under particular circumstances. We shall explore possible ways of formulating the process of rational choice in situations where we wish to take explicit account of the "internal" as well as the "external" constraints that define the problem of rationality for the organism. [Simon-55]

Led by Simon's ongoing efforts (see [Simon-82]), many economists explored various sorts of bounded rationality models, and these sorts of investigations remain an active area of study in economics and operations research. See, for example, [Arthur-91], [Tietz-92], [Honkapohja-93], [Wall-93], [Norman-Shimer-94], and [Lilly-94]. Unfortunately, we do not have time to survey this interesting and relevant literature here.

Economics and operations research have had close ties with decision theorists since von Neumann and Morganstern [vonNeumann-Morganstern-44] introduced decision theory. While economists were exploring bounded rationality, decision theorists were becoming aware of the difficulties caused by their theories' not taking into account the cost of thinking.

Savage [Savage-67] was one such decision theorist, but he feared that repairing the theory might lead to paradox. Let us look at how this might happen. Standard decision theory says that a rational agent chooses the optimal action, given its preferences and beliefs (this may not be the "truly" optimal action, since the agent's beliefs may be inaccurate). In a limited rationality version of the theory, we might say that the agent is not powerful enough to process its beliefs and preferences completely, so instead we have some model of the limitations on the agent's computational capabilities, and we say that the agent chooses optimally, given those additional limitations. But this seems to be begging the question: if we already know the agent is incapable of making optimal decisions, doesn't this conflict with the assumption that it makes optimal use of its limited computational capabilities? Is this not somehow contradictory?

In [Hacking-67], Hacking responds to Savage's concerns. Hacking proposes that the problem with decision theory is due to the property of logical closure[1] among beliefs in the theory of personal probability used by Savage. Hacking points out that the probability axioms can be expressed in terms of "possibility," and that the theory itself, and all the traditional arguments in support of it, are really independent of the exact definition of possibility that is used. Possibility, in turn, is reduced to knowledge (the idea being that p is possible if it is not known to be false). And finally, Hacking proposes an "examiner's sense of knowledge," in which the objects of knowledge are *sentences* in some "personal language," rather than propositions. Using propositions as the objects of knowledge leads to logical closure because logically equivalent propositions are standardly defined to be identical. In contrast, with Hacking's examiner's view of knowledge,

> ...a man can know how to use modus ponens, can know that the rule is valid, can know p, and can know $p \supset q$, and yet not know q, simply because he has not thought of putting them together. ([Hacking-67], p. 319)

This definition of knowledge leads to what Hacking calls "personal possibility," and to his preferred, sentential form of personal probability. This solves Savage's problem in one

1. Logical closure on beliefs is referred to as "logical omniscience" in the logic community (such as [Hintikka-75]), according to Konolige [Konolige-86].

sense, by making the theory weaker. Now that agents do not make all logical deductions, to restore some of the power of decision theory we would need to model what deductions they *do* make.

But Hacking realized this still left a question unanswered: namely, how are we supposing the agent decides what deductions to make? (This corresponds directly to the question of how a game playing program decides what partial game-tree to examine, which we explored earlier.) Hacking points out that if we don't want to assume an agent acts optimally at the base-level of choosing an action, then we also shouldn't assume it acts optimally at the meta-level of deciding what to deduce—making an optimal decision at that level might be just as expensive, or more, than making the deductions needed for an optimal base-level decision. So it appears that an agent will have to also do some work to "police" its own meta-level deductions—but then, does it not require yet more reasoning to do this "police work," and more reasoning to police the police work? And so on? To get out of Savage's "paradox," would we not have to incur an "infinite regress?[1]"

Hacking does not worry about the infinite regress problem, saying that he is satisfied to just say that the metareasoning stops at some arbitrary level and just makes a fast approximation to the optimal answer. This may be a pragmatic view, but it is a somewhat dissatisfying answer from a theoretical perspective, because it is not clear what a claim of limited rationality would consist of if it did not require that *something* be optimized—either the decision of what to do, or the meta-decision of what computations to do to decide what to do, or the meta-meta-decision, *et cetera*. But at the same time, it appears that any such optimality requirement would require unlimited reasoning powers on the agent's part to assure that it is satisfied. With limited resources, is there any kind of optimality assumption that makes sense at all?

This question was recently answered in the affirmative by Lipman [Lipman-91]. Lipman showed that it is indeed consistent to simultaneously say of an agent both that it is uncertain about the outcomes of its possible options at every level, and that it chooses the optimal action, within the bounds of that uncertainty. This amounts to saying that an infi-

1. Economists have long been wary of this so-called "infinite regress" problem with regard to bounded rationality—Lipman suggests [Winter-75] as an introduction to that literature.

nite tower of levels of suboptimal meta-decisions can be treated as an optimal decision among alternatives at a single level.

However, the model Lipman uses for his proof is very simple (there is no mention of probabilities, for example), and is not intended to be a complete theory of rationality in the tradition of decision theory. It also does not tell us how to find an optimal decision, it is merely an existence proof that there is some optimal thing that an agent can do despite its being limited on all levels. Nevertheless, it is a landmark work and contains many interesting ideas that will undoubtedly be important for future limited rationality research.

Next we will look at some of the limited rationality frameworks that have been developed in the context of computer game playing.

3.2 Efforts Within Computer Game-Playing

As I discussed in chapter 2, the problem of limited rationality occurs in a very concrete form in efforts to write programs to play games like chess. In principle, players could just make optimal moves by examining the entire game tree and applying the minimax rule (see §2.4.1). But in practice, this is infeasible, and no easier way to choose optimal moves is known.[1] Thus, programs have traditionally adopted a strategy of approximating optimal play by doing only a partial search. This now raises the issue of control of search, which was traditionally handled in games by doing a flat search. Unsuccessful attempts at search control using "forward pruning" techniques (see [Truscott-81]) were abandoned early on. Later, some *ad hoc* "singular extension" mechanisms specialized for Chess were explored ([Anantharaman-et-al-90], [Hsu-et-al-90]). But more interestingly, as we saw, researchers began to explore general techniques for exploring those parts of the tree thought to be most important for making a good decision ([Rivest-88], [McAllester-88], [Schaeffer-91]). However, these authors were not concerned with the general problem of limited rationality, and did not express what their algorithms were doing in decision-theoretic terms.

In chapter 2 we briefly mentioned Russell and Wefald's CGP work. We now return to examine their work in more detail, because of its intent, unlike most other CGP work, to

1. Except in the few games that have been solved analytically, such as Nim (see [Berlekamp-et-al-82]).

address general issues of limited rationality, in addition to CGP applications in particular. Russell and Wefald's work was the original inspiration for the investigations of rational decision analysis on which I report in the next section.

In a series of papers and a book ([Russell-Wefald-89], [Russell-90], [Russell-Wefald-91], and [Russell-Wefald-91a]), Stuart Russell and Eric Wefald set forth their view of the problem of limited rationality as seen from a CGP perspective, and presented their solution to the problem, in the form of a series of game tree search algorithms.

More important than their particular algorithms, however, is their general characterization of limited rationality, which we will now briefly describe. In their framework, actions that are available to an agent are separated into *external* actions and *computational* actions. Both kinds of actions may affect the current world-state. The world-state is imagined to be divided into external and internal portions; computational actions will generally directly affect only the internal portions, whereas the agent's utility directly depends only on the external portions. The value of a computational action is defined by reference to its effect on the agent's choice of external action.

Russell and Wefald's general characterization of the metareasoning problem focuses on the possible *computations*, i.e., sequences of computational actions, that might precede the next external action. The value of the computation is the difference in value of the actions that would have been chosen before and after the computation.

However, in practice they break down the problem into an incremental form by only considering, at any given time, which computational action to perform next, and making the simplifying assumption (the *single-step* assumption) that the next external action will take place immediately afterwards. (Thus the computations being considered always have a length of 1 step.)

Those are the basic concepts of Russell and Wefald's characterization of limited rationality.

3.3 Application to Decision-Analysis

Expert decision-making systems often evaluate decision trees[1] to determine what action has the best expected utility. But sometimes the decision tree implied by the system's model of the domain is intractably large; not only too large to store explicitly, but also too large to evaluate in a reasonable time. System designers typically avoid such difficulties by simplifying the model of the domain so that it will not product large decision trees, or by abandoning regular decision trees when they become too large and using some other representation to support the decision analysis, such as Markov trees [Moskowitz-84], [Wong-et-al-90], [Sonnenberg-Beck-93].

An alternative, explored here, is to partially evaluate the decision tree, using estimates for the utilities of non-leaf nodes. This is analogous to the standard computer game-playing technique of partially evaluating the typical intractably-large game trees. (In fact, the notion of a *single-player game with chance*, while not the primary focus of CGP research, subsumes decision-tree-based decision analysis in general.)

The problem now (as in game-tree search) becomes two: estimating utilities of non-leaf nodes, and exploring those parts of the tree that are expected to contribute the most to the quality of the overall decision. (See §2.3 for a general discussion of this way of breaking down the problem.)

In CGP, the problem of tree search can be partially handled using pruning algorithms such as Alpha-Beta and Scout.[1] Recently, as we have seen, researchers have gone further, using methods that use information about the degree of uncertainty of utility estimates to select those search actions that will most improve the quality of the overall decision. These methods have met with some success. Especially appropriate are the decision-theoretic metareasoning techniques of Russell and Wefald and similar methods by Baum and Smith, which search those parts of the tree expected to contribute the most to the utility of the final decision. Can the same sort of decision-theoretic metareasoning be profitably used to control the partial evaluation of decision-analysis trees? In my research I have begun to explore that possibility. This section will report on the progress made so far.

---

1. The term "decision tree" is used in decision analysis to refer to a tree of possible future decisions and their nondeterministic outcomes. This is to be contrasted with the usage of the term "decision tree" in machine learning contexts, where it typically means a classification tree of criteria used to categorize different problem instances (for example, in ID3-like systems). In other areas of AI the term "decision tree" is used in still other ways (cf. [Winston-92]). In this thesis we will restrict our use of the term to the decision-analysis context.

1. We have not discussed Scout in this paper. See [Pearl-84] for a detailed description and proof of optimality, and [Bodkin-92a] for a correction of the error in the proof.

3.3.1 Initial investigations

The goal of these investigations is to determine whether decision-theoretic metareasoning, as conceived by Russell and Wefald, could be applied to large decision-analysis problems as well as to two-player perfect-information games, where they applied it. It was noticed that in Baum and Smith's version of decision-theoretic search control, they mention the possibility of allowing chance nodes in their game trees, although the algorithm for this case was not fully developed. However, it seemed likely that the algorithm could be extended to include the chance nodes, and eliminate the second player, reducing the case to the traditional one in decision theory: that of a single person against chance or nature.

I decided that these investigations would require a concrete decision-making domain, within which to conceptualize the details of the algorithm and experiment with an actual implementation. I chose the game of Spider Solitaire for this, because it represents an intractably-large decision-analysis problem, and yet it is easily approachable for research due to its structure being defined by a relatively simple set of rules, as in most games. For more discussion of the motivation for using games, see §2.1 (p. 18).

Once decision-theoretic metareasoning is implemented for this domain, our task will be to compare it with other approaches experimentally, and see whether the metareasoning provides any benefit, in terms of achieving higher average utilities. We may also consider the effect of time, and judge whether higher utilities are still achieved when the time cost is taken into account.

Either a positive or negative result would be interesting. If the result is positive, this opens the door for future work that attempts to apply metareasoning to harder decision problems, perhaps even more "real-world" problems, such as large medical decision-analysis problems (e.g., [Wong-et-al-90]).

If the result is negative, then this tells us something about the limits of the Russell-Wefald limited rationality approach, and may point the way towards generalizations or extensions of the technique that might make it more generally applicable. Or, we may discover some fundamental limitations to this entire style of metareasoning which prevents it from being appropriate for a large class of search problems.

3.3.2 Motivation for Spider Solitaire

I chose Solitaire as my example of a decision-analysis problem for some of the same reasons that many AI researchers have chosen games as a problem domain in general. (See §2.1 for a discussion.) However, the model in decision theory is not typically that of a reasoner and a diametrically-opposed opponent, as is the case in two-player games, but rather is that of a reasoner against nature, a "chance" opponent that is imagined to make moves at random according to some fixed probability distribution. Thus the natural gaming equivalent of decision-theoretic situations is the single-player game with chance. Typical examples are dice games and single-player card games.

Why are single-player card games like dice games? A more natural way to think of card games is as games with "imperfect information," that is, the game has some true state at any time, and the effect of actions is deterministic given this state, but part of the information about the state is hidden from and unknown by one or more of the players. In multi-player card games such as Poker and Bridge, different players have different information, and indeed, in such games, the effect of an action that gains more information (such as drawing new cards) can not be adequately modeled as a chance event with definite probabilities, because due to the fact that different players have different knowledge, they will have different probabilities as to what the cards drawn will be. However, if there is some information about the state that is unknown to all the players (such as the identity of cards yet to be drawn from a deck), the revealing of that information can be taken as a chance event with probabilities determined by the probabilities that would be assessed by a hypothetical observer who knew everything that any player knew. However, since none of these players know what these probabilities would be, it does not pay for them to view the game in this way.

However, in a single-player card game such as a Solitaire game, if there is any hidden information at all, it is of course hidden to all players, and thus when new information is revealed, e.g., cards are turned over, this can be adequately modeled as being chosen randomly from the remaining unseen cards at the time-of-revealing instead of at the time of the last shuffle of the deck. (This makes the uncertainty appear incrementally; whereas reasoning about all the possible permutations the entire deck might occupy would be overwhelming.) Thus, Solitaires with hidden cards can be viewed as nondeterministic games where "nature" or "chance" alternates moves with the human or computer reasoning agent. So a decision tree is an appropriate model of these games.

Many Solitaire card games exist and are described in various books of rules of games (e.g., [Morehead-68]). A number of interesting variations (such as "Seahaven Towers") involve no hidden information at all, and the reasoning situation is just that of a deterministic puzzle, like Rubik's cube or the 15-puzzle, where the initial state is given randomly, but all the operators available have deterministic outcomes, and the problem is to plan a sequence of actions that will, with certainty, achieve the goal. Such puzzles are not so interesting from my viewpoint; I am concerned with reasoning situations that are similar to the kinds of real-world situations where goals typically cannot be achieved with certainty, but more often, only with high probability. I am interested in the problem of figuring out how to maximize your expected utility, and in the metareasoning problem of determining what information to gather to best help one make this decision. Thus, I confine my interest for the present to Solitaires with hidden cards, although ultimately I would like to convert some of my techniques to the perfect-information case as well.

There are also many Solitaires that have hidden cards. Unfortunately, many of them are not very interesting as reasoning problems because there is so little information available to the player, and so little flexibility in one's actions, that it seems that even an optimal strategy would not perform very much better than a relatively naive one. ("Klondike" solitaire is one example.) Thus, in such domains, there would not be much feedback in terms of performance, and large numbers of games would have to be played to determine the relative ordering of two programs' average performance with any statistical significance. Thus, I looked for a game that offered more opportunities for good reasoning to affect the outcome.

The Solitaire game "Spider" is such a game. As Woods says in [Woods-89], "Spider is a particularly challenging double-deck solitaire. Unlike most solitaires, it provides extraordinary opportunities for the skillful player to overcome bad luck in the deal by means of careful analysis and complex manipulations." This is due to the fact that Spider offers many possible ways to arrange existing cards before new ones are revealed, making it more likely that some arrangement will be found that will have a high expected utility upon the revealing of the new information. If not as many moves were possible, it would be more likely that no move would provide a significantly higher expected utility than any of the others. Thus Spider presents the possibility that appropriate use of decision analysis may lead to good performance.

3.3.2.1 Evaluating the method

To apply decision theory to the problem of playing Spider, we require a utility function. I chose to define the utility of a Spider game via a somewhat arbitrary scoring function of the final position achieved.[1] The function is borrowed from a common spider program, and seems to be a good measure of "closeness" to a winning position—i.e., if

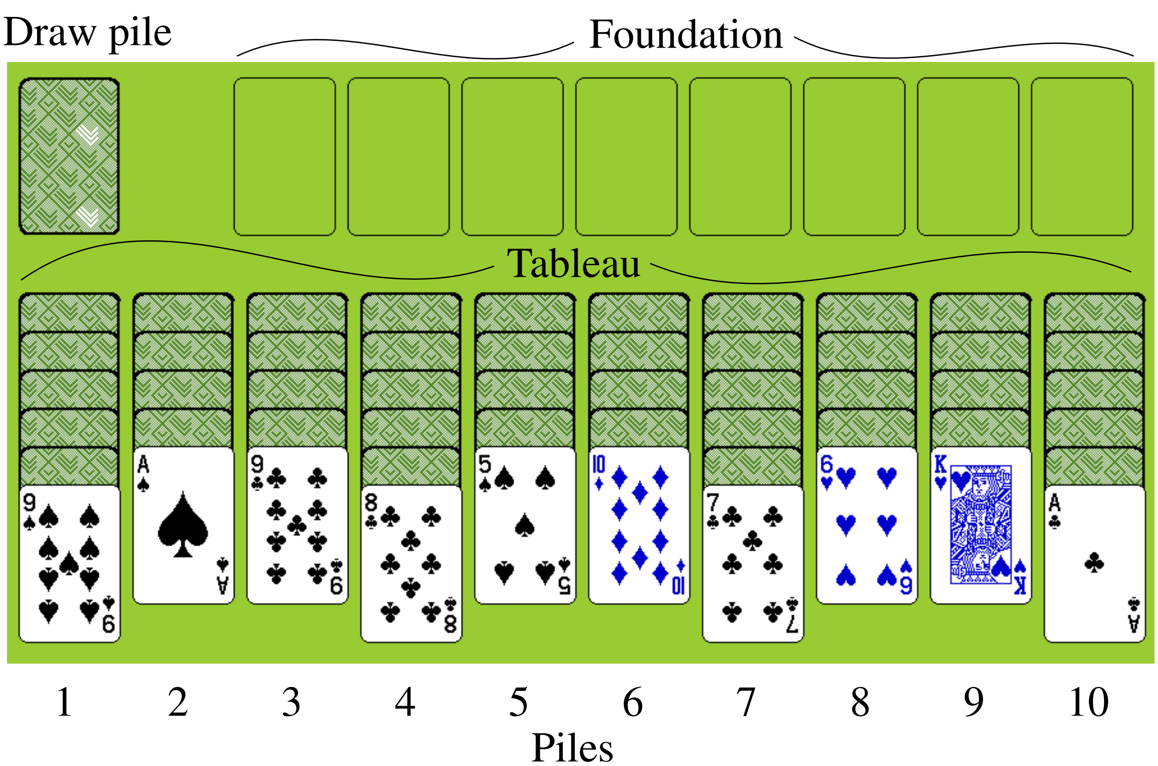

Figure 10. Example of an opening position in spider solitaire.
The moves available in this position are:
Naturals: 8♣→9♣, 7♣→8♣
Others: 9♣→10♦, 8♣→9♠, 6♥→7♣, 5♠→6♥

1. There is an interesting analogy between this measure and the "Quality-Adjusted Life-Years" utility measure sometimes used in medicine (e.g., in [Pliskin-et-al-80], [Perlman-Jonsen-85], [Tsevat-et-al-88], [Hogenhuis-et-al-92], [McNeil-et-al]), which we do not have time to delve into here.

you have a higher average score, then you probably actually win the game more often as well. (Statistics could be collected on this once players are implemented.) The score is defined in any game-state, which allows it to be used as an evaluation function as well. Except in rare cases it is monotonically increasing as the game progresses, so it can serve as a lower bound on the final score that will be achieved. It is thought that this measure will allow "greedy" approaches to be fairly successful. Thus, if a program performs better than a greedy algorithm, it will be doing pretty well.

Using such a scoring function rather than just the win/loss status of the game also allows learning algorithms to have a chance if they start out never winning any games. (In Spider, a random player, for all practical purposes, never wins.)

Our investigations will begin with the construction of various random players. The point of the random player is to establish a minimal baseline such that if we notice that some technique is doing no better than random, we know that the technique is worthless. We will then establish a series of progressively better but still-simple players so that we can compare our final metareasoning player to those as well.

We will generally collect data in the form of a joint distribution over score achieved versus time taken. From this data we can extract many different measures of performance: average score achieved per game, average time per game, average total utility when time incurs an additive linear cost (the amount of which would be an additional parameter), and rate of accumulation of benefits (scores) over time.[1] We can compare the different algorithms under these different measures, giving us more information about the algorithms than if we were to just collect a single evaluative number for each one.

3.3.3 Results to date

In the attempt to characterize reasoning about Spider as a partial search of a decision-analysis tree, and apply Russell-Wefald and Baum-Smith methods to that tree, several problematic issues arose. The first realization was that the nodes in our tree could not represent the true state of all the cards, because that state is unknown. So instead, we

1. This measure is discussed in a machine-learning context by Rivest and Sloan in [Rivest-Sloan-93]. It leads to a significantly different view of the problem, one that is perhaps not so appropriate for medical decision-making. Also one can view Allis's proof-number search [Allis-et-al-91] in this way.

decided to fix the level of our states to be that of “information states” describing all that we know about the state at a given point—all the visible cards. As we mentioned earlier, it is possible to treat the outcome of turning over a new card as if it were a chance event, rather than the gathering of information about the true unknown state. This fits better into the standard decision-analysis model, where we know the outcomes of all the chance events that have occurred so far, but not the outcomes of future ones. (A detailed discussion of how to treat incomplete-information games in terms of “information sets” is currently taking place in the CGP community. See [Smith-Nau-93], [Blair-et-al-92], [Blair-et-al-93], [vanLent-Mutchler-93], and [Gordon-93].)

So using the chance model, Spider is exactly a decision-tree problem. However, when considered in this way, the branching factor is very large: every time one turns over a new card, it may generally be any of however many of the 52 different kinds of cards haven’t had both their instances turned over yet. (There are two of each card initially.) In the early parts of the game, this is still close to 52. And if we really want to do a proper decision analysis, we have to consider all the branches, because our utility (and evaluation function) could potentially be different in each one. (The probabilities, however, can only have one of two values, depending on whether 1 or 2 instances of the given card remain.)

This forces us to face the problem that a partial search of the Spider decision tree will not get very far. And it seems to be the wrong thing to do. Often in Spider, we make moves that turn over new cards, but we don’t care right away what the identities of those new cards are—we may have revealed the card simply as a side-effect of moving some other card in some sequence of moves. But the decision-tree paradigm forces us to separately analyze each possible outcome of the card-revealing “experiment,” even if that card’s identity is completely irrelevant to the next few moves we might make.

One can try to fix this problem by mutilating the concept of a “move” in Spider: you can break a card-revealing move into two parts: the part that uncovers a card, making it available to be revealed, and the part that turns over the card to reveal it. Normally in Spider these are considered one move, because there is never any harm in gaining information earlier rather than later. But breaking it into two allows us to postpone the high-branching-factor card-revealing moves, pushing them farther down into our decision tree, below the part of the tree we will examine in our partial search. This allows our partial search to encompass many more “tactical”-style moves consisting of reorganizations of cards that we could do if we had immediately reveal the new card and repeat our search below that node for each possibility of what the card might be revealed to be.

But playing with the definitions of moves in the game seems like the wrong thing. As human players, we know we can always turn over the exposed cards right away, because getting the information about their identity earlier rather than later never hurts. However, at the same time, when we are *thinking* about a sequence of moves, we are quite capable of ignoring or “abstracting out” the identity of the revealed cards, and thinking about the possible transitions between *abstract* states consisting of specifications for the identity of some of the revealed cards, but not of others.

So we could go ahead and say that we will fix the level of abstraction to be one where the identity of some cards may be unspecified, and declare that a possible move from such a state is to “look” at one of the abstracted-over cards, and incur at that point the branching chance event that allows us to consider separately the different possibilities for that new card.

But now that we are already thinking in terms of searching through abstractions, it becomes clear that a fixed level of abstraction is not sufficient, because there are many other kinds of abstractions about spider moves that would be useful as well. For example, sometimes we want to dissect our thinking about the identity of the new card into just two cases (instead of 52)—is it a King (for example), or is it not a King? Those might be the two natural possibilities to consider in a particular situation when planning what to do after the card is revealed. Another example: often there are two sequences of moves that will leave you in the same board position (transpositions). We want to be able to think about what to do after such as sequence without having to repeat the reasoning for every possible way of getting to the position. (This has long been recognized in Chess and is handled with transposition tables—see [Newborn-89]—and it leads to the exploration of a game graph instead of a tree). But even beyond that: sometimes there are two moves available that can easily be proven to be strategically equivalent, in that, for example, two piles of cards can be switched without changing our description of the known part of the state. The obvious way to handle this abstraction is to canonicalize our pile order so that all states that are identical under such a transformation are treated as a single (more abstract) state.

We can go even farther. In Spider, many moves are reversible, so that a set of positions that is spanned by a graph of reversible links can be thought of as a single, more abstract position, with moves leading from it to other groups of positions that are all connected by reversible moves.

We could, indeed, write a Spider program that includes special-case code to handle all these different ways of abstracting over Spider positions, and write more and more code as new opportunities for abstraction become apparent. However, in this enterprise we would seem to be doing inherently the wrong thing. Wouldn't it be better to generalize all these kinds of abstractions and deal with them all uniformly, instead of making some *ad hoc*, spider-specialized code for each one? For, certainly, we will find similar (but not identical) kinds of abstractions that would be appropriate for other games, and we would like our work on Spider to carry over to those.

So, after investigating these issues, it became clear that a straightforward decision-analysis of Spider would not address the really interesting issues that come up when thinking about how to write a program to play the game. In fact, because of the problem of the large branching factor, a straightforward decision-analysis, would probably perform very poorly compared to a special-purpose Spider program that took advantage of these abstractions, no matter how good the search-control was—simply because in the flat state-space view of Spider, the states are not abstract enough.

Although it would be possible to go ahead and implement the program and perform the experiments on the metareasoning technique, I felt that due to the above realizations, I had already achieved a result with regard to the general applicability of the game-tree search control techniques: namely the realization that those techniques were all designed around a state-space search paradigm, which simply was not appropriate when dealing with a game where many kinds of abstractions are apparent and important at many levels, and the sorts of reasoning that are natural and effective in that domain involve constantly shifting around to different levels of abstraction.

In the next chapter we will discuss how reasoning with abstractions might be dealt with in a way that might be amenable to adapting some of the state-space search control techniques to work with abstractions instead.

In summary, the result of our initial investigations on reapplying LR techniques developed for game-playing to a decision-analysis problem is that the techniques are not directly applicable—even, interestingly enough, when the decision-analysis problem studied was, actually, a game. This points out that fact that even within computer game-playing, the focus of the existing CGP work has primarily been within a very restricted class of games, namely the two-player, perfect-information ones, and that it is surprisingly difficult to reapply the existing CGP techniques to other sorts of formally-definable games.

However, the possibility is left open that with a more intensive effort, CGP techniques, especially the ones using general decision-theoretic metareasoning principles such as Russell and Wefald's, might still be adaptable to different kinds of search domains.

3.4 Other LR work in AI

So our initial investigations into the reapplication of CGP limited rationality techniques to decision analysis were somewhat unsuccessful, although they led to an interesting alternative direction for research. However, it is worth mentioning related efforts in AI to apply LR ideas to practical problems.

Horvitz and his colleagues have long been investigating the problem of how to deal with limited resources in medical decision-making and other contexts. Much interesting work which we do not have time to survey here can be found in their papers; see [Horvitz-88a], [Horvitz-et-al-89a], [Horvitz-et-al-89b], [Horvitz-Breese-88], [Horvitz-89], [Horvitz-et-al-89c], [Breese-Horvitz-90], [Heckerman-et-al-89]. Like myself, Horvitz is applying meta-level control to decision analysis. However, Horvitz's focus is on controlling the refinement of the probabilities associated with the arcs or transitions in a decision tree, whereas my interests lie in refining the estimated or expected utilities at the leaves of a partial decision tree, via deepening the tree in places where the accuracy of those estimations is most in need of refinement.

In the planning community, there has been much work on meta-level control and the LR issues that naturally arise there. The work by Dean and his colleagues is especially central in this area, but again we do not have time to survey it in detail. See [Dean-Boddy-88], [Boddy-Dean-89], [Dean-et-al-93a], [Dean-et-al-93b], [Dean-et-al-92], [Dean-91]. Much of Dean's work utilizes the concept of *anytime* algorithms that can be halted at any time to return an answer of less than optimal quality. Shlomo Zilberstein ([Zilberstein-93], [Zilberstein-Russell-93]) has continued in this line of investigation by examining how to optimally combine anytime subprograms to achieve good overall limited rational behavior.

Recently, Russell and his students have been extending Russell and Wefald's earlier work on general LR theory by formalizing a concept of *bounded optimal* agents ([Russell-et-al-93]). Although it is quite interesting, and will undoubtedly be a good foundation for future work on formal LR, it still falls into a highly state-oriented category of formalisms; whereas my interests lie more in definitions of LR that do not assume a state-space ontology. I will now discuss my ideas along those lines.

3.5 Towards a formal theory

Frameworks for limited rationality such as described in [Russell-et-al-93] are a good start, but they tend to be closely tied to the state-space search paradigm, which as I have discussed, is really inadequate—ultimately we need, instead, to search through a space of possibilities expressed at varying levels of abstraction.

Could we possibly develop a theoretical framework for limited rationality that *doesn't* assume a definite level of state-space structure? One in which time, and world-states, and actions, are really infinitely subdivisible, and it is only the epistemology of our agents, their language of thought, that is discrete? Perhaps this will not turn out to be feasible, but at least, we can conduct some preliminary theoretical explorations in an attempt to find out.

3.5.1 Motivation

We need a theoretical framework for describing *limited rational agents*, agents that are designed with the goal of performing as well as possible, given a fixed level of computational power, when time is a factor in performance.[1] Such agents are in contrast to agents that are designed to find optimal solutions to problems, regardless of how much computing time will be consumed. As described earlier, Russell and Wefald have argued persuasively in support of the limited-rationality approach, and they have begun to develop theoretical tools for formalizing what limited rationality means, and for explaining the behavior of limited rational agents.

In this section we begin to build the foundations for a somewhat more general (and, we think, ultimately more useful) theoretical framework for describing limited rationality, and we show how various concepts defined in the Russell-Wefald framework, and other frameworks, can be expressed within the new framework.

3.5.2 Probability notation

There are many formalizations of probability theory in use today. Although the differences between them are largely just unimportant mutations of notational syntax, we feel that in order to achieve complete clarity when talking about probabilities, it helps to set forth the exact, formal system and notation that we will be using, rather than just assume that all readers are familiar with the particular syntax being used. This will also give us opportunities to define some convenient, abbreviated notations for certain probability concepts that we will use especially often.

Our probabilities, events, and random variables will be defined over a space Ω of outcomes (sometimes called a *sample space*). For now, our definitions will require that Ω be finite; the infinite case requires more care in the definitions, but in either case the theoretical tools for limited rationality that we will build from them will be used in the same way.

Definition 1: A *distribution* p is a function $p: \Omega \to [0, 1]$ from the outcome space Ω to the real interval from 0 to 1, subject to the standard probability-theory constraint that

1. Of course, some entities have the option of increasing their computational power, in which case they are faced with a broader trade-off, between making better use of their computational resources, or paying the costs of amassing more. For example, an AI research group may buy new computers instead of improving their algorithms, or the industrial economy as a whole may choose to develop faster semiconductors, rather than finance extensive research into limited rationality theory.

$$\sum_{\omega \in \Omega} p(\omega) = 1. \tag{23}$$

We treat a probability distribution as a function rather than as an operator (compare with the more-standard $P\{\omega\}$ notation) because we will want to consider many different distributions $p, q, r, \ldots$ computed by different agents at different times.

Definition 2: An *event* $E \subseteq \Omega$ is a set of outcomes.

Although distributions are function of outcomes, it will also be convenient to define the result of applying a distribution to an event, as follows:

Definition 3: Let p be a distribution, $E \subseteq \Omega$ an event. Define $p(E)$ to be $\sum_{\omega \in E} p(\omega)$.

The following is the standard definition of conditional probability.

Definition 4: Let p be a distribution, and let E and C be events. Then define the *conditional probability of E given C under p*, written $p(E|C)$, to be $p(C \cap E)/p(C)$, i.e., the probability that the event C occurs and also the event E occurs, divided by the total probability that the event C (called the *condition*) occurs (regardless of whether E occurs or not). If $p(C) = 0$ then $p(E|C)$ is undefined.

Unfortunately, there is no standard, concise way to denote, for a given condition Y, the corresponding function mapping events to their conditional probabilities given Y (although this function can be denoted using lambda calculus as $\lambda X.p(X|Y)$). We now introduce a convenient notation for this concept:

Definition 5: Let p be a distribution, C an event. Then we define p_C to designate the distribution *p given C*, defined as

$$p_C(\omega) = \begin{cases} 0 & \text{if } \omega \notin C \\ p(\omega)/p(C) & \text{if } \omega \in C \end{cases}. \tag{24}$$

Note that if $p(C) = 0$, then p_C is a partial function; in fact, it is undefined everywhere. Using this definition together with definition 3, we can designate the conditional probability of event E given event C under distribution p as simply $p_C(E)$, where p_C is the desired function that maps events to their conditional probabilities.

Definition 6: A *random variable* V is a partial function $V: \Omega \to R$ from outcomes into some range R.

We do not restrict R to be a numeric range as in some other formalisms; however, certain operations (for example, the expectation operator, definition 8) over random variables will only be defined in the case where R is numeric.

We also define a new notation for the event that a random variable applied to the outcome has a particular value.

Definition 7: Let V be a random variable, and let $x \in \mathrm{range}(V)$. Then the expression $V = x$ denotes the event $\{\omega \mid V(\omega)\mathrm{defined} \wedge V(\omega)=x\}$.

Definition 8: Let V be a *real random variable*, that is, a random variable with a real-valued range, and let p be a distribution. Then the *expectation of V under p,* written $\mathrm{Ex}_p[V]$, is given by:

$$\sum_{(\omega \in \Omega) : V(\omega)\mathrm{defined}} p(\omega) V(\omega). \tag{25}$$

Additionally, it will also be convenient for us to define the meaning of applying a real random variable to an event, subject to a distribution:

Definition 9: Let V be a real random variable, E an event, p a distribution. Then $V_p(E)$ will denote $\mathrm{Ex}_{p_E}[V]$, the expectation of V under p given E

The point of this definition is that we will later define utility just to be some real random variable U whose expectation we happen to be trying to maximize, and so definition 9 will automatically let us express the utility of an event E (subject to a particular distribution p) by $U_p(E)$.

We also define a special notation for building random variables out of other random variables:

Definition 10: Let $V_1, V_2, \ldots, V_n$ be random variables, and let $e(v_1, v_2, \ldots, v_n)$ be an expression containing $v_1, v_2, \ldots, v_n$ as terms. Then $\langle e([V_1], [V_2], \ldots, [V_n])\rangle$ will denote a random variable defined as follows:

$$\langle e([V_1], [V_2], \ldots, [V_n])\rangle(\omega) = e(V_1(\omega), V_2(\omega), \ldots, V_n(\omega)). \qquad (26)$$

For example: Let X be a real random variable. Then $\langle \log [X] \rangle$ denotes a random variable whose value for a given outcome is the logarithm of X's value for that outcome.

3.5.3 Domain Languages, Information States, and Agents

Our goal in this chapter is to provide a formalism for talking about limited rational agents. To talk about an agent's rationality, we need to have some way of describing its what the agent is supposed to be doing, what domain it is working in.

We will assume there is some set W of the possible world-histories of the agent's domain. We will use W as the outcome space for talking about probability distributions. We also assume there is some language L, called the *domain language*, for talking about the domain in which the agent is working (sometimes called its *micro-world*). The language L is a set of sentences that make assertions about the domain. Each sentence $s \in L$ has some content, namely, the set of possible world-histories of the domain that are consistent with the sentence's being true, i.e., the event that the sentence is true. Let $M: L \to 2^W$ be the meaning function mapping sentences to their content.

We note that for agents that do metareasoning, the problem domain concerns not only the part of the world that is external to the agent, but also the internal computations of the agent itself. Thus some sentences in L may make assertions about the agent's internals.

Definition 11: An *information state* of an agent is a set of sentences $S \subseteq L$. Intuitively, an information state represents the information held by the agent at some time.

This is not to say that an agent necessarily represents its information as a list of sentences; just that we can describe an agent's information in this way.

Definition 12: The *content* of an information state S, written $M(S)$, is the intersection of the contents of all the sentences in S:

$$M(S) = \bigcap_{s \in S} M(s). \tag{27}$$

Thus, the content of an information state is an event as well; the event that all the pieces of information (sentences) in the information state are correct.

Definition 13: Let X and Y be information states or sentences. We say X entails Y (written $X \to Y$) iff $M(X) \supseteq M(Y)$.

We do not assume that an agent necessarily has access to the content of any information states or sentences, and we do not assume that the agent is aware of any particular entailment relations between them. The agent's information state is not assumed to be closed under entailment.

Generally, an agent will gain information over time, which will mean that the size of its information state will increase, and thus the size of the content of its information state will decrease, because fewer world-histories will be consistent with the agent's increased knowledge.

Now, all this is not to say that the agent will actually store its knowledge by explicitly listing the set of world-histories in its information state. Rather, this is just a theoretical tool that will be useful for explaining what a limited rational agent is doing, particularly one that works by deliberative metareasoning, i.e., by explicitly considering different acts of reasoning that might be performed. We are not here talking about what kind of knowledge an agent might have or how that knowledge is represented, we are just saying that whatever an agent's state of knowledge, there is a corresponding event.

Similarly, for each action that an agent may perform, there will be a corresponding event—the event that the action is performed, the set of world-histories in which the action is performed.

Most formalizations of agents, such as Russell and Wefald's in [Russell-Wefald-91a], give the domain more structure than just saying it is a set of possible histories. Namely, they say that the world of the domain outside of the agent is always in some discrete world-state, and that the world passes through a sequence of states in response to an agent's actions. This kind of structure will be important for us later, but for now we can proceed without it, and just talk about events.

3.5.4 Rational Interpretations of Agents

A rational interpretation of an agent is a way of ascribing beliefs and values (probabilities and utilities) to the agent that makes all its decisions appear to be rational. Under a rational interpretation of an agent, the agent is always acting so as to maximize the expectation of its utility under its distribution.

Formally, we will say that, relative to the outcome space Ω and domain language L, a *rational interpretation* R of an agent α is a pair (p, u) of a distribution p over Ω, and a random variable u over Ω, such that whenever α is in information state $S \subseteq L$ and is required to immediately (without changing its information state) choose between mutually-exclusive actions $a_1, \ldots, a_n$, it will choose an a_i that maximizes $u_{p_{M(S)}}(E_i)$, where $E_i \subseteq \Omega$ is the event that action A_i is the one chosen.

Thus, we will say that for any limited rational agent, there is, implicit in its design, some distribution p over Ω, and a random variable u over Ω, and that they have the following property: when the agent is in knowledge state K, and it is forced to immediately commit to one of a set of mutually-exclusive actions $A_1, \ldots, A_n$, it will commit to the A_i that maximizes $u_{p_{M(K)}}(A_i)$, i.e., the one with maximum expectation of u under p given K. When we observe the agent committing to an action A, we explain this by saying that p and u must be such that the utility of A under p given K is higher than that of the other actions to which the agent might commit itself.

As analyzers of a particular rational agent, we may not know exactly what the functions p and u are, and the agent itself will not in general know exactly what they are, in the sense of being able to give, for any ω, the values $p(\omega)$ and $u(\omega)$. But we surmise that the p and u exist, because that is useful for our model. If the agent is deliberative in its rationality, it may be able to report *some* information about p and u, namely, that information about p and u that it is explicitly computing and using.

To emphasize again: we are not presuming that an agent actually chooses actions by computing the sum of the product of probability and utility over all the world-histories in the event of that action being performed. Nothing of the sort. Rather the story is as follows.

Someone gives us an agent. They claim it behaves rationally. What does this mean? This means it behaves rationally relative to some p and u. Which p and u? We may not know exactly, but we can deduce constraints on p and u by observing what decisions the agent actually makes.

For an agent that is claimed to perform deliberative reasoning, we can do more: it will work explicitly with some probabilities and utilities, and when it does, it will enable us to deduce extra information about the functions p and u.

3.5.5 Limited Rational Agents

Relative to a fixed W, L, and M, we say that an agent α is a limited rational agent, if, for any rational interpretation $I = (p, u)$ of α, α fails to take full advantage of its information when constructing its probabilities and utilities. For example, in a limited rational agent, it may be the case that its p assigns nonzero probability to some world-histories that an agent with unlimited computational power could prove were impossible.

The key to handling limited rationality within standard decision theory is to accept that a limited rational agent will assign some nonzero probabilities to some impossible world-histories. That is, we do not assume that our outcome space ω consists only of possible world-histories; it may include impossible ones as well, and p may assign them nonzero probability. For example, suppose the agent is about to perform a certain deterministic computation. There is only one possible outcome. But in advance of the computation our agent will not know what the outcome will be. Thus our agent can be seen as assigning nonzero probability to an impossible outcome, namely that the computation comes out differently than it actually will.

3.5.6 Notation for describing metareasoners

In our framework, we will be describing various agents that explicitly compute various probabilities and utilities for use in their decision-making. For each agent, we will interpret what it is doing as follows:

For each agent α, we imagine that there is some distribution p_α which is the distribution over outcomes that is implicit in the design of α. Now, the agent α, having limited resources, cannot possibly know or compute its probability $p_\alpha(\omega)$ for every possible $\omega \subseteq \Omega$, because the space of possible world-histories Ω will generally be much too large. Instead, α will generally be computing the probability $p_\alpha(E)$ of some event $E \subseteq \Omega$, or even the conditional probability $(p_\alpha)_B(A)$ of an event A given another event B. Or perhaps the agent is computing the expected value $\mathrm{Ex}_{p_\alpha}[E]$ for some event E. But rather than requiring that the agent compute such values constructively from $p_\alpha(\omega)$ values, we *interpret* the values the algorithm computes as having meanings such as the above, and we surmise there is some p_α "built into" the agent that is consistent with all the computed values. The computed values can be seen as empirical data placing constraints on what the agent's "actual" distribution p_α might be. In other words, we are using probabilities as a way of explaining and interpreting what the agent is doing; we are not providing rules for how the agent must compute its probabilities. The method of computation will vary among the different reasoning algorithms.

Now, what of change over time? Generally, an agent's beliefs will change over time. One way to handle this is by indexing the distribution by both the agent and by the time: $p_{\alpha, t}$. However, this does not provide us with any structure relating our agent's distribution at one time with its distribution at another time. A better way is to imagine that for each agent there is but one distribution p_α, but that, after our agent discovers new information, then when we talk about the probability of an event E, we are implicitly referring to the conditional probability of E given the event I that expresses all the information that α has discovered so far, i.e., we mean $(p_\alpha)_I(E)$.

Now, moreover, we will treat computations themselves as information-gathering operations just like those that gather information about the outside world, as do Russell and Wefald. So when we perform some computation that enables us to assess a more refined value for the probability of some event E, this does not mean that $p_\alpha(E)$ has changed, rather it means the following: let C be the event that the particular computation in question came out in the particular way that it did. Before the computation, when we referred to the probability of E, this meant just $(p_\alpha)_I(E)$, where I is the event representing all the information we had gathered so far. After the computation, when we refer to the probability of E, it means $(p_\alpha)_{I \cap C}(E)$, because we have now also gained the information that C holds. It is consistent to have different values for $(p_\alpha)_I(E)$ and for $(p_\alpha)_{I \cap C}(E)$, at least, it is if $(p_\alpha)_I(C) \neq 1$. Generally, we do not expect the agent's distribution p_α to contain perfect information about what the results of all our computations will be. The results of all our computations could in principle be figured out by a reasoner with unlimited resources, but p_α is not intended to model the outcome probabilities that an ideal reasoner would assess, rather it represents the probabilities implicit in the design of our limited rational agent, who is unable to compute in advance what the results of all his future computations might be.

This perspective on probability and computation allows us to formally make sense of various concepts that metareasoning researchers have heretofore used informally. For example, Baum and Smith's agent uses distributions over what its probabilities will be in the future after doing more computation. In particular, they have, for leaf nodes in their game tree, distributions over what the probability of winning will be assessed to be at that node after that node is expanded and "more search" is done below it.

In my framework I can formalize this concept in terms of the following entities:

- p, the distribution for our agent
- I, the event expressing whatever information we currently have.
- N, the event that a given game-tree node (board position) is reached in the actual game.
- W, the event that we win the game.
- E, the event that we indeed expand the node in question before making our next move.
- C, a random variable for all the computational events that will occur before we make our next move. In other words, for a given world-history ω, if $\omega \in I$

(i.e., if ω is consistent with what we know so far), then $C(\omega)$ is defined to be the event expressing all the information that we know, in that history, at the time the next move is made. For $\omega \notin I$, we leave $C(\omega)$ undefined.

Given these entities, we can express our current value for the node by just $p_{I \cap N}(W)$—the probability of winning, given everything we know so far and given that we do eventually move to the node in question.

Now, using definition 10, we can define a random variable V for Baum and Smith's notion of "the future probability of winning that we will assess after gathering more information," as follows:

$$V = \langle p_{[C] \cap N}(W) \rangle. \tag{28}$$

V is the desired function mapping world-histories to the probability of winning we assess for the node at the time of our next move, in that world-history. Note that V is a partial function that is undefined wherever C is, namely, for those world-histories outside of I. This is okay since we only care about world-histories that are consistent with what we know so far.

So when Baum and Smith assess their distributions over future values of nodes, they are computing values of the function $\lambda x . p_{I \cap E}(V \Rightarrow x)$. That is, a distribution over the future value of the node, given everything we know and given that the node is expanded.

What I mean is that when a Baum-Smith agent assesses such a distribution, we can understand and formally describe what it is doing in that way.

3.5.7 Knowledge-relative utilities

Decision analysts have found that when people are required to assess their utilities for different outcomes, they do not give one consistent set of values; their assessments change from one moment to the next [Tversky-Kahneman-87]. We could attempt to explain this phenomenon by saying that people do not report their true utilities accurately, and instead their reports are noisy readings of their true utilities. However, experiments intended to measure utilities directly using lotteries showed the same effects. I think that a fairly obvious introspection shows that we really do change our evaluations of possibilities from one moment to the next, and that this effect is not just a difficulty in reporting the "true" utilities presumed to lie inside us. The reason we change our utilities is also obvious, namely that as time passes we think of more ramifications and consequences of the outcome in question, which in real life is never really a total, final outcome, or a complete world-history, but instead always has future effects that bear contemplation.

However, one may choose to model the situation by saying that for each individual there *are* definite utilities that the individual would arrive at if he were suddenly to become omniscient, or to have infinite reasoning and computation capabilities, or some such, and the utilities an individual reports are simply his estimates, based on the information and thinking he has attained so far, of what *his true* inner utility for the proposed event would be.

But for my purposes, I would like to pursue a different approach. It seems implausible to me that what a person reports as his values are a mere estimate of that person's true inner values that neither that person nor anyone else can ever arrive at. It seems to turn a person's moment-to-moment values into a mere epiphenomenon, when in reality they are all that we can ever really deal with. So instead, I propose to treat a person's moment-to-moment values as the fundamental, primitive entities in my further discussions of utility. How, then, are a person's changes in values to be explained? Well, I propose that a person's *utility structure* be a function from possible states of knowledge to utility functions over outcomes, which give the utilities that the reasoner would assess for the outcome in that state of knowledge. (If our model of the reasoner is not exact, we might also wish to generalize this to be a probability distribution over utility functions, where the actual utility function in a given knowledge-state is chosen at random according to the distribution.) Formally:

Definition 14: Let S be a universe of possible states of knowledge, L a language of outcomes, and α an agent. We write $U(\alpha)$ for *the utility structure of* α, which is a function giving, for each knowledge state $s \in S$, the *utility function of* α *in* s, written $U(\alpha)(s)$, which is a function from outcome descriptors $l \in L$ to utility values $U(\alpha)(s)(l) \in \mathbf{R}$.

So, we imagine that although α cannot change its utility structure, it remains true that since α's state of knowledge s is constantly changing, the current utility function $U(\alpha)(s)$ may in general change as well.

This gives us a framework in which we can describe particular types of agents, such as "rational" agents, that change their utilities in particular ways when they obtain new information. Our descriptions of types of agents can now be formalized as restrictions on the utility structure $U(\alpha)$.

This approach is incompletely developed, but it perhaps points at some useful directions for future work.

3.6 Directions for future development

It is still unclear which of the above ideas concerning LR formalisms will prove most fruitful. I think that we have not quite arrived yet at the appropriate foundation on which to build an LR theory. However, in the above efforts we are getting closer to success. One qualification is that we need to give more thought to deciding exactly what will be the nature of the entities that will be the objects of belief and utility. For example, we have already decided that we do not wish to assign utilities only to states at single moments in time; assigning them to world-histories would be more flexible. However, when we consider the limited nature of our agents, we can see that focusing on concrete, completely-specified histories as the primitive objects of our agent's beliefs is assuming too much; the agent can really only deal on a level of partial, incomplete *descriptions* of sets of outcomes, and can only assign probabilities and utilities to those. Perhaps we want to think of these descriptions as something like Konolige's *epistemic alternatives* [Konolige-86], in that they do not correspond to possible worlds, or "impossible possible worlds," rather they correspond to worlds that are possible *as far as the agent knows*. This hearkens back to the concepts of *personal possibility* discussed by Hacking ([Hacking-67]). Also, they should perhaps be used in a way similar to Lipman's use of states, in which the set of states can always extended to encompass a larger number of more refined states that take into account additional considerations about the world.

So what are these objects we are trying to get at, exactly? Are they syntactic entities? We perhaps do not want to assume or attach a particular language to them; we may want them to have identities apart from the particular language we choose to express them. However, it is difficult to see exactly how to do this. So instead of saying what the epistemic alternatives are themselves, perhaps we should just say that we are just always working with some set L of unique *labels* $l \in L$ for each of our epistemic alternatives (hereafter EA's), whatever they are. This is similar to how some treatments of thermal physics avoid discussion of what states really are by working with sets of labels for states instead. We do not care what these labels are, or what the language is like that we use to label the epistemic alternatives. However, in contrast to the case in physics, we want our epistemic alternatives to correspond to abstractions, in that some may be more or less specific than others; some may subsume, or be implied by, others. We could do this by mapping each EA to the set of world-histories to which it corresponds, and then using the subset relation, but again, we are trying to get away from even talking about a fixed level of concrete, completely-specified things. So to be more flexible, perhaps we can just say that there is some partial ordering on L that expresses the subsumption relationship. Of course our agent will not in general know this ordering, i.e., it is not true that for all pairs of labels (l_1, l_2), he knows whether the EA labeled by l_1 entails the EA labeled by l_2. The various entailment propositions, and their negations, may even be epistemic alternatives themselves.

Armed with our concept of epistemic alternatives, we can rephrase all the definitions presented in the previous subsections in terms of them, as opposed to the existing explication based on sets of world-histories. We can have probabilities and utilities for each of the epistemic alternatives, indexed by the agent and the time of reasoning, and whatever else we want. We can interpret various EA's as being explicitly considered by an agent when it constructs internal representations for them, similarly to how we considered a game state to be "examined" when an agent constructed a representation for it. The explicit EA's might have a special status in our description of what in agent is doing when it is reasoning.

However, a theory based on such a foundation might be too weak. Also, in these efforts we are a long way from doing anything that will effectively constrain our search for good, practical, limited-rationality architectures and algorithms. But let us reemphasize that this work is of a very exploratory nature; we are examining a wide range of possible theoretical foundations in hope of someday finding a better, more general theory of rationality. To do this, we are considering abandoning some of the traditional restrictions on the nature of the objects of belief and utility (i.e., that they are propositions, or sentences, or complete world-states, or sets of such), and in doing this, we are perhaps ignoring some of the motivations that led to those restrictions originally. But a priori, we cannot know that those restrictions will not lead to unrealism or inaccuracies in a theory of reasoning based on them. Indeed, we saw that treating belief as concerning propositions can lead to an unrealistic consequence, namely logical omniscience.

But the more generalized concept (of epistemic alternative) that we are examining here, while perhaps being general enough to build just about any model of reasoning upon, has not so far constrained us to the point of finding what the good models are. To make further progress, we will perhaps need to experiment with more specific and constraining theories, even if they might be considered arbitrary, to see if they will get us somewhere.

At a broader level, one might ask whether perhaps in psychology and the social sciences this sort of exploration will ever work—perhaps good models are inherently harder to find in those fields than in physics, say, because the important concepts in those realms have a very "fuzzy" nature (whereas the important quantities and entities in physics have a more clearly definitive nature).

On the other hand, just because success in rationality theory (and other areas related to theoretical AI) is difficult doesn't mean we should give up. The benefits from explorations like this could be huge. We are, after all, a wealthy society (compared to those that existed in earlier eras) and can afford to spend some time seeing whether techniques of mathematical modeling that worked well in, e.g., physics will also work well for the perhaps less clear-cut matters of agents, societies, economies, beliefs, preferences, rationality, and so forth.

Now, in the next chapter we will go on to look more carefully at some of the abstraction issues that were suggested by our investigations of how to apply limited rationality techniques to Spider solitaire.