Project Description

1. A Fictional Example

The sun shone brightly overhead, but the light was strangely muted, like dusk on the high desert of Utah. The red cliffs towered overhead, and the coral pink sand spread to the horizon. Despite the sun, the sky was a dark dark blue, with the stars shining through. There was a flicker of motion in the foreground, and Oscar crawled into view, dragging himself torturously over the rocky surface. Oscar was in trouble.

The day had begun splendidly, as Oscar undertook a circumnavigation of the large mesa. But two thirds of the way around, while exploring a side canyon, Oscar tried to climb over a rock pile that was less stable than it appeared, and he found himself sliding uncontrollably down the sandy slope to end wedged between and under some of the newly dislodged rocks. After working for some time he managed to wrench himself free, but as he did he heard the heartrending scream of tearing metal and realized that he had hurt himself. He quickly realized that his solar panel was no longer functioning. Oscar relied upon that panel to quench his voracious thirst as he explored the Martian desert. He soon realized that he had also broken his radio antenna, and was no longer able to communicate with the Lander. And as he rolled forward across the dune, it became apparent that he did not have full power in his right rear tractor assembly.

The problem was serious. The sun was dropping rapidly towards the mesa, and Oscar had to get back to the Lander before it was too dark to see. Oscar had been on Mars for a week, and in that time he had learned that parts of the surface formed a loose powdery quicksand. If he got into that he would flounder, and it required considerable power to extricate himself. Power and time were the two things he could not afford to waste. He had only his limited battery power to get him back to the Lander, and the route back was through new territory. He could not return the way he had come and make it back before dark, and if he were caught in the cold Martian night the power needed to maintain minimal operating temperature would drain his battery and leave him to spend eternity on the floor of the Martian desert.

Oscar had to strike out towards the Lander along the most direct route. Fortunately, Oscar had learned to detect distant fields of quicksand by their color and texture, enabling him to plan a route around them. Of course, he was not infallible at this. Sometimes he would find himself surrounded on three sides by quicksand he did not detect earlier and have to backtrack. This time, he could not afford much backtracking. To make matters worse, as the Martian dusk settled upon him, it became increasingly difficult to make out colors and textures. Oscar was equipped with flood lights that would illuminate his path, but they changed appearances and made it harder to judge the surface, and their continued use would drain his battery.

As Oscar picked his way laboriously through the rocks and over the sand, a new threat emerged. In the west he could see an ominous haze that foreshadowed one of the howling Martian sandstorms. In the thin atmosphere, 200 mph winds could drive the sand into every joint and rapidly disable machinery. Oscar knew that, faced with the sandstorm, the Lander would be tempted to use its limited mobility to take refuge in one of the canyons, but without radio contact it could not tell Oscar that it was doing so. They had a previous arrangement regarding where the Lander would go in an emergency, just to take care of the possibility of radio failure. But they had not foreseen the simultaneous failure of the solar panels. Oscar could either make his way to the landing site or to the canyon, but if he guessed wrong about where the Lander was he would not have enough power to make up for his mistake.

Fortunately, on the basis of what he had learned during his week on Mars, Oscar was able to make a reasonable prediction of how long it would be before the sandstorm reached the Lander, and estimate that it was more probable than not that the Lander would be waiting for him at the landing site when he arrived. Indeed, it was, Oscar was able to plug into the Lander's power supply and replenish his battery as they jointly took shelter from the storm, and later he was able to repair his solar panel. The Fourth Martian Expedition ended happily.

Oscar was from a new breed of rovers endowed with sophisticated epistemic capabilities. His earliest brethren had been lunar rovers that relied upon direct control from Earth to get them out of difficulties. The one minute time lag it took radio signals to travel from Earth to the Moon meant that these rovers could not respond instantly to unforeseen threats, and many of these early rovers succumbed to natural disasters, like falling into meteor craters when the lunar surface suddenly and unexpected crumbled beneath their weight. The earthbound scientists realized that this problem would be magnified intolerably on Mars, where at the best of times radio signals would take 15

minutes rather than one minute. Their initial attempt to solve this problem was to endow Mars rovers with sophisticated planning capabilities. Unfortunately, this did not lead to the construction of survivable rovers. The problem was that planning must be based upon both general knowledge and specific knowledge of one's immediate surroundings. Conditions on the surface of Mars were known only imprecisely — why else explore it? Scientists on earth could equip rovers with their best understanding of Martian conditions, but these could not possibly cover all contingencies, and they were bound to be wrong sometimes. It was imperative that a rover be able to learn for itself about the dangers of the Martian environment. To do this, it needed sophisticated epistemic capabilities.

Faced with these problems, one NASA scientist suggested that they should look into the philosophical literature on epistemology. But Dr. A replied, "Epistemology? That is philosophy! We're scientists — we don't do philosophy." His colleague, Dr. B, chimed in, "Yeah, we'll solve this problem by just writing a program for knowledge acquisition." But then A observed, "Hmm, but what exactly do we want the program to do? I think we need a better specification of the design goals." B suggested, "I know — let's ask a psychologist". But the reply they got was, "Oh, we psychologists don't study good reasoning. We study the reasoning people actually do, be it good or bad. We are often more interested in all the stupid things people do. If you want to know how good reasoning works, you need a theory of rationality. That is the domain of the philosopher." And so, happily, NASA turned back to the philosophical literature and learned all about defeasible reasoning and its application to perception, temporal projection, causal reasoning, the discovery and application of probabilities, etc.

2. The Need for Sophisticated Cognition

I devoted as much space as I did to the example because I often get the reaction that AI does not need solutions to the problems I aim to solve — AI systems do not require such sophisticated cognitive abilities. The example of the Mars rover is an argument to the contrary. Any sufficiently sophisticated cognitive agent must draw conclusions on the basis of sensory information and make decisions about how to act on the basis of those conclusions and its background knowledge. Some of the background knowledge can be precompiled, but much of it will have to be acquired by the agent in its working environment.

If our aspirations are sufficiently limited, hacking may suffice to produce a system that satisfies our needs. But to build a truly sophisticated system of information processing, we need a general characterization of what the system is supposed to do. This will both aid us in designing the system and enable us to confirm that the finished system meets our design goals. The design goal is to have a system that is capable of acquiring the kinds of information it needs, and do so in such a way that the conclusions it draws are reasonable conclusions to draw given its input. A general characterization of what conclusions are reasonable is a theory of rationality. The construction of a truly sophisticated cognitive agent must presuppose a theory of rationality. That is largely a philosophical theory.

Scientists in other disciplines are sometimes inclined to dismiss philosophy as airy-fairy speculation, but that just demonstrates an ignorance of what goes on in philosophy. At least those parts of philosophy that are of relevance to building cognitive agents are precise and highly mathematical. In fact, all theoretical work in AI is based on philosophical preconceptions, although generally rather simple (often simplistic) ones. For example, numerous AI systems are based on results in formal logic (e.g., the completeness of resolution refutation). Formal logic sits squarely on the border between philosophy and mathematics. As the logic gets more sophisticated (e.g., incorporating modalities) the study of it tends to become focused more in philosophy than in mathematics. Similarly, most work on decision-theoretic planning is based on classical decision theory, which, although subsequently embraced by economics, was originally a philosophical theory about rational action. My general point is that even simple AI systems are based upon philosophical theories, although in many cases the theories have become so entrenched in other disciplines that their origins may go unrecognized.

Although existing AI systems are based upon philosophical theories concerning rationality, it is my contention that they often make incorrect use of them, and further that the theories themselves are sometimes wrong and that the best way to correct them is to see how they lead us astray in building cognitive agents. There should be a synergistic relationship between philosophy and AI. The research proposed here is aimed at making sophisticated use of theories of rational cognition in the construction of cognitive agents, and using our observations about the performance of such agents to refine or correct the theories of rational cognition, thus improving the agent design. Let us look at this in more detail.

3. Epistemic Cognition

Consider the kinds of epistemological capabilities we want a sophisticated epistemic cognizer to have. I will illustrate these by focusing on the example of the Mars rover.

Perceptual Judgments To make routine judgments about its current situation on the basis of sensor input, the rover needs to engage in perceptual reasoning. An agent that is cognitively less sophisticated than a human being could simply read the output of its perceptual system (an image) as a veridical account of its surroundings. This is the approach adopted by a lot of contemporary work in robotics. However, human cognition does not regard the image as the epistemological endproduct. If the agent believes the image to be veridical, it will take its surroundings to be the way they appear, but sometimes the agent will bring other non-perceptual knowledge to bear and judge that the image is not a veridical representation of its surroundings. For example, in trying to decide how firm the ground is before it, it is desirable for our rover to be able to decide that the ground is not as red as it appears (redness being an indication of softness we can suppose) because it is being viewed in the light of the setting sun. In other words, the image provides only a defeasible reason for judging that the environment is as represented in the image, and a sophisticated agent should be able to defeat that defeasible presumption and arrive at conflicting beliefs in some cases. To reason in this way, general principles of defeasible reasoning from perceptual images must be implemented in the rover. A preliminary implementation of these principles is presented in [37]. It is based on the OSCAR system of defeasible reasoning, which will be discussed further below. A deeper discussion of these principles can be found in [48].

Temporal Projection Inferences based upon current perception can provide the rover with some knowledge of its surroundings. However, that is of little use unless the rover can also draw conclusions about its current surroundings on the basis of earlier (at least fairly recent) perception. For instance, suppose the rover wants to judge whether the sand to the right is redder (and hence probably softer) than the sand to the left. We suppose the rover has the ability to make a visual judgment of the redness of the sand it is currently looking at. But as the rover cannot look at both patches of sand at the same time, it will not be able to make the comparison using only current perception. The rover can look at one patch and draw a conclusion about its redness, but when the rover turns to look at the other patch, it no longer has a percept of the first and so is no longer in a position to hold a justified belief about how red it is *now*. This is a reflection of the fact that perception *samples* bits and pieces of the world at disparate times, and a cognitive agent must be supplied with cognitive faculties enabling it to build a coherent picture of the world out of those bits and pieces. What the rover needs is some basis for believing that the first patch of sand has not changed color in the brief interval since it was inspected. In other words, the robot must have some basis for regarding the color as a *stable property* — one that tends not to change quickly over time. This is provided by a defeasible principle of *temporal projection*. As a first approximation, such a principle might have the form:

If $t_0 < t_1$, believing *P-at-*t_0 is a defeasible reason for the agent to believe *P-at-*t_1.

In [37], I defended a more sophisticated version of this principle and discussed an implementation of it in OSCAR. It is clear, I think, that some such principle of defeasible inference must be included in the epistemology of our rover.

Causal Reasoning Obviously, the rover should be able to reason about the causal consequences of its actions and the causal consequences of exogenous events that it witnesses. This requires, among other things, a solution to the frame problem. Various solutions have been proposed. I favor the solution I proposed in [37]. Again, my proposed solution has been implemented within OSCAR. It is noteworthy that most of the solutions that have been proposed in the literature presuppose some form of defeasible or nonmonotonic reasoning.

Reasoning about Probabilities Most of the rover's reasoning about its current situation will be based in part on its general beliefs about its environment. Most of these general beliefs will assign conditional probabilities to various eventualities. A certain amount of this probabilistic information can be supplied by the mission designers, but if the world being explored is sufficiently unknown to warrant exploration, then there must be a lot of antecedently unknown probabilities. The rover must have the ability to discover, for example, that the color and texture of the sand is a probabilistic indicator of its softness. Section six will look more carefully at the epistemological problem of acquiring the desired probabilistic knowledge.

Building an autonomous rover requires taking these problems seriously and implementing solutions to them. You cannot solve the problems without engaging in epistemological analysis.

4. The OSCAR Architecture

One reaction I get from AI researchers who are not familiar with my work is that it is all very nice to speculate about how such sophisticated cognition works, but it will be many years before we are in a position to build implemented systems with such capabilities, and in the meantime we need to concentrate on solving problems that are within our grasp. What these people are unaware of is just how much I have already accomplished along these lines. My research has produced the OSCAR architecture for cognitive agents, and preliminary implementations of all of the above varieties of reasoning are currently running in OSCAR. (The currently disseminated version of OSCAR can be downloaded from my website.) As will become apparent below, there is still much work to be done, but the goal of building robots capable of such cognition is not distant fantasy.

Defeasible Reasoning The current version of the OSCAR architecture is described in [32]. The core of the architecture is a system of defeasible (non-monotonic) reasoning. Defeasible reasoning is reasoning that makes the conclusion reasonable without guaranteeing its truth deductively. The previous section illustrates that sophisticated cognition makes heavy use of defeasible reasoning — a conclusion that is now universally accepted in philosophical epistemology. Defeasible reasoning has been investigated in both AI and philosophy. Originally, neither group of researchers was aware of the other, and the work was done independently, but now there is considerable crosstalk between the two research cultures. I was one of the first philosophers to argue for the importance of defeasible reasoning, writing about it first in my dissertation in 1965, and then in numerous later articles and books [20,21,22,23,24,25,26,27,30,31,32,33,34,35,36,37,48,38,41,47].

A defeasible reasoner has two parts. First, it produces arguments for conclusions, some of which can be defeaters for steps in other arguments. Second, given a set of such arguments, it applies an evaluation algorithm to determine which conclusions to adopt in the face of that set of arguments. These are the conclusions supported by undefeated arguments. I will call such conclusions *justified*. So justification is relative to a set of arguments. Given a particular set of inputs to the system, we can consider the set of *all* the arguments that can be constructed starting from those inputs, and define a *warranted* conclusion to be one that is justified relative to that maximal set of arguments.

An evaluation algorithm computes the set of conclusions that are justified according to some semantics for defeasible reasoning. A number of different semantics have been proposed, the best known of which are versions of circumscription and versions of default logic. I produced my first semantics for defeasible reasoning in 1979, but only published it in full in 1986 [27]. It was later shown [49] to be "almost equivalent" to Dung's preferred model semantics [6]. However, I discovered some intuitive counterexamples to that semantics, and proposed a different semantics in 1995. This has only recently been shown [57] to be equivalent to the currently popular stable model semantics of Bondarenko et al [1]. Recent work on decision-theoretic planning led me to the realization that my 1995 semantics was still inadequate for some purposes, and I published a revised semantics in [46]. I will talk about that further below.

Most implemented defeasible reasoners work only for very impoverished languages in which validity is decidable (i.e., recursive). Typically, they only work for some version of the propositional calculus. But such languages have inadequate expressive power to provide the knowledge representation machinery for a sophisticated cognitive agent. OSCAR is the only implemented defeasible reasoner that works in a rich logical language like that of first-order logic. The source of the difficulty here is that, as Israel [9] and Reiter [51] showed long ago, the set of warranted conclusions for a defeasible reasoner is not generally going to be recursively enumerable. Familiar systems of automated deductive reasoning produce recursively enumerable sets of theorems, and so cannot be generalized straightforwardly to defeasible reasoning. However, I solved the problems of how to build a general-purpose defeasible reasoner that avoids this difficulty in my [30] (see also my [32]). I showed that if the set of reason-schemas and defeaters used by the reasoner satisfy certain reasonable constraints, then the set of warranted conclusions is Δ_2 in the hyperarithmetic hierarchy. That has the consequence that it is possible to build a reasoner with the following two properties:

(1) If a conclusion is warranted, the reasoner will eventually reach a stage at which the conclusion becomes justified and no further reasoning will render it unjustified.

(2) If a conclusion is unwarranted, the reasoner will eventually reach a stage at which the conclusion is unjustified and no further reasoning will render it justified.

In other words, for each conclusion, the reasoning will eventually stabilize so that warranted conclusions are justified and unwarranted conclusions are unjustified.

Unfortunately, there can never be a guarantee that the reasoner has reached the point where the status of a particular conclusion has stabilized. This illustrates the fact that human reasoning is defeasible in two different senses. First, it is *synchronically defeasible*, in the sense that adding premises can change what conclusions are warranted. This is the kind of defeasibility studied in non-monotonic logic. But human reasoning is also *diachronically defeasible*, in the sense that even without adding premises, further reasoning can change the defeat status of a conclusion. This is a very important feature of human reasoning. Our reasoning is defeasible in the sense that we adopt conclusions tentatively, with the understanding that we might have to take them back later. What is important here is that we really do adopt the conclusions, even if the adoption is tentative. In a rich and undecidable language for knowledge representation, reasoning is non-terminating. But agents have to act. They cannot wait for the end of a non-terminating process before deciding how to act, so they act on the basis of the currently justified conclusions. This is what humans do, and this is what artificial agents can do as well if they are based on a defeasible reasoner that computes the set of justified conclusions as it goes along and changes the justification status later if necessary. This is the way OSCAR works.

Natural Deduction In addition to an algorithm for computing justification, an implemented defeasible reasoner must construct arguments. The general form of OSCAR's reasoning is analogous to that of a *natural deduction* theorem prover. Most theorem provers in AI reduce formulas to clausal form and then reason by resolution refutation. By contrast, OSCAR reasons with formulas in their full "natural form". Natural deduction theorem provers are distinguished by two characteristics. First, they can reason by "making suppositions for the sake of the argument", drawing conclusions relative to those suppositions, and then employing inference rules that enable them to *discharge* the suppositions. For instance, natural deduction theorem provers generally employ a discharge rule to the effect that if the conclusion Q has been inferred relative to the supposition P, the reasoner can conclude $(P \rightarrow Q)$ independently of the supposition. Second, natural deduction theorem provers reason bidirectionally, reasoning backwards from the conclusion sought as well as forwards from the given premises. The first version of OSCAR's deductive theorem proving was described in [29], and again in [32]. A more detailed description can be found in *The OSCAR Manual*, which is available on my website.

OSCAR's natural deduction theorem prover turns out to be extremely efficient. For example, a problem that has often been used as a test problem for automated theorem provers is the "Schubert steamroller problem":

$(\forall x)(Wx \rightarrow Ax)$ $\quad$ $(\forall x)(\forall y)[(Cx \ \& \ By) \rightarrow Mxy]$
$(\forall x)(Fx \rightarrow Ax)$ $\quad$ $(\forall x)(\forall y)[(Sx \ \& \ By) \rightarrow Mxy]$
$(\forall x)(Bx \rightarrow Ax)$ $\quad$ $(\forall x)(\forall y)[(Bx \ \& \ Fy) \rightarrow Mxy]$
$(\forall x)(Cx \rightarrow Ax)$ $\quad$ $(\forall x)(\forall y)[(Fx \ \& \ Wy) \rightarrow Mxy]$
$(\forall x)(Sx \rightarrow Ax)$ $\quad$ $(\forall x)(\forall y)[(Wx \ \& \ Fy) \rightarrow \sim Exy]$
$(\exists w)Ww$ $\quad$ $(\forall x)(\forall y)[(Wx \ \& \ Gy) \rightarrow \sim Exy]$
$(\exists f)Ff$ $\quad$ $(\forall x)(\forall y)[(Bx \ \& \ Cy) \rightarrow Exy]$
$(\exists b)Bb$ $\quad$ $(\forall x)(\forall y)[(Bx \ \& \ Sy) \rightarrow \sim Exy]$
$(\exists c)Cc$ $\quad$ $(\forall x)[Cx \rightarrow (\exists y)(Py \ \& \ Exy)]$
$(\exists s)Ss$ $\quad$ $(\forall x)[Sx \rightarrow (\exists y)(Py \ \& \ Exy)]$
$(\exists g)Gg$ $\quad$ $(\forall x)(Gx \rightarrow Px)$
$(\forall x)[Ax \rightarrow [(\forall w)(Pw \rightarrow Exw) \rightarrow (\forall y)((Ay \ \& \ (Myx \ \& \ (\exists z)(Pz \ \& \ Eyz))) \rightarrow Exy)]]$

$(\exists x)(\exists y)[(Ax \ \& \ Ay) \ \& \ (\exists z)[Exy \ \& \ (Gz \ \& \ Eyz)]]$

This is a slightly whimsical symbolization of the following:

> Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants. Therefore, there is an animal that likes to eat a grain-eating animal. [[19], 203]

The current version of OSCAR solves this problem by making 471 inferences, 145 of which are used in the final proof. By this measure, the reasoning is 28% efficient. Theorem provers based on resolution refutation tend to have much more difficulty with this problem, making thousands of inferences and ranging between .02% efficient and 9%

efficient [32] (page 162). On easier problems, OSCAR's efficiency tends to range between 75% and 100%. (See the sample problems on my website.)

A couple of years ago, Geoff Sutcliffe, librarian for the TPTP library ("Thousands of problems for theorem provers") proposed a "shootout" between OSCAR and Otter, a well-regarded resolution refutation theorem prover. Sutcliffe selected 163 problems from the TPTP library, and OSCAR and Otter were run on the same machine. Of the problems that Otter got, OSCAR failed to get three. Of the problems that OSCAR got, Otter failed to get 16. Of the problems that both got, OSCAR was on the average 40 times faster than Otter, despite the fact that OSCAR is written in LISP and Otter is written in C. So OSCAR is a fast and efficient theorem prover.

Large Databases A common problem for AI systems is that they may be able to solve a problem when given just the relevant information, but when given lots of additional irrelevant information the proof search bogs down and they become unable to solve the problem. The difficulty is one of being unable to retrieve the right information at the right time. This will be a serious problem for an autonomous robot operating in the real world, because it must store information relevant to all the problems it may encounter. This difficulty has been addressed in OSCAR in a preliminary way. My general hypothesis that there are two keys to solving the relevance problem. First, the reasoning has to be efficient, in just the sense illustrated above. OSCAR's bidirectional reasoning does relatively little redundant search. Second, information storage must be done efficiently. This is done in OSCAR by storing information in a syntactically based discrimination net. I have done one very preliminary experiment to test this hypothesis. I gave OSCAR six fairly difficult deductive reasoning problems (the Schubert Steamroller problem was among them), and then added 100,000 irrelevant premises. It turns out that this slowed OSCAR down by only a factor of 2. It is my impression that OSCAR is the only automated deductive reasoner that can solve such artificially bloated problems. It is worth noting that the same strategy cannot be used for resolution refutation theorem provers, because the discrimination net requires that formulas retain their full first-order form and not be reduced to clausal form.

It must be emphasized that this is just one preliminary test. I want to improve the technique, and test it on large causal reasoning and planning problems. In the latter connection, a classical planner has been implemented within OSCAR that works by reasoning defeasibly about plans. The defeasibility comes in via the search for threats. It is not a particularly impressive planner, being roughly comparable to UCPOP, but it should suffice for this kind of test. (note that the currently popular satisfiability planners and MDP planners don't have a ghost of a chance of solving such large problems.)

Further Comparisons It is easy to compare OSCAR's deductive reasoning to other systems, because there are numerous published results from other automated deductive reasoners. However, OSCAR's principal strength is that it can perform general-purpose defeasible reasoning. I cannot compare OSCAR to other defeasible reasoners generally, because there are none that can solve the full range of problems OSCAR finds trivial. (For examples, see the file "PC-examples.lisp" on my website.) There has, however, been considerable work on implementing solutions to some specific defeasible reasoning problems, mainly those connected with the frame problem and the Yale Shooting Problem [7],[8]. OSCAR solves the Yale Shooting Problem in 4 milliseconds on a rather slow Macintosh G4 (see my [37] for details). For this purpose, the problem is encoded by giving OSCAR the following premises:

```
(the_gun_is_loaded at 20)
 ((Jones is alive) at 20)
 (the_gun_is_fired at 30)
 (the_gun_is_fired when the_gun_is_loaded is causally sufficient for
                ~(Jones is alive) after an interval 10)
```

and the query

```
(? ((Jones is alive) at 50))
```

There are other systems that purport to solve the Yale Shooting problem. The monotonic approaches stemming from the work of Lin & Reiter [14] have spawned a lot of research, but they seem unpromising for the reasoning of agents in domains of real-world complexity. They proceed by adopting "explanation closure axioms". The axioms used might be true in toy problems, but are certainly not true in the real world. For example, the Yale Shooting Problem is solved by adding an axiom to the effect that the only way for Jones to cease to be alive is for him to be shot. In the real world this would have to be replaced by an axiom enumerating all possible ways of terminating Jones' life. The problem is that we simply don't know what all the ways of killing someone are, and if we did the disjunction of them would make the axiom so long as to be completely unmanageable.

Other than OSCAR, the most fully developed defeasible approach to reasoning about causes involves circum-

scription and employs either some version of the situation calculus or the event calculus. Shanahan [56] has a nice review of this work up to 1997. My reaction to this work is two-fold. First, it is "baroque". It seems that each causal reasoning problem has to be coded anew, and the axiomatizations employed are incredibly complex. Second, they proceed by reasoning deductively from the circumscription of the resulting theory. But the circumscription is a second-order theory, and second-order reasoning cannot be fully implemented because it follows from Gödel's theorem that there is no complete proof algorithm for second-order logic. Again, this does not seem like a promising way of designing a practical real-time agent. Compare it with OSCAR, which encodes problems very simply and reasons about them very quickly.

Just to mention one other example, Shoham {67} proposed the "extended prediction problem", which he conjectured no automated system would be able to solve. It is a problem involving colliding billiard balls. OSCAR solves the problem in 0.4 seconds on the same G4 Macintosh. (See my [37] for details.) For this purpose, nothing sophisticated was done to accelerate the spatial reasoning. No doubt, this could be made much faster. Some more sample problems can be found on my website, along with LISP code for the currently disseminated version of OSCAR.

The upshot is that OSCAR is fast — easily fast enough to provide the inference engine for a real-time agent — and OSCAR is able to handle both deductive reasoning problems and defeasible reasoning problems that other systems are either unable to solve or can handle only with considerable difficulty.

5. Defeasible Reasoning

If OSCAR is such a great reasoner, why do I need support? Unfortunately, OSCAR is not perfect. Minimally, a semantics for defeasible reasoning should tell us what conclusions are justified given a set of arguments supporting these and other conclusions some of which may be defeaters for steps in some of the arguments. Viewed in this way, the currently disseminated version of OSCAR is based on a semantics that is equivalent to the currently popular stable model semantics [6].

However, there is a more general problem of which this is only a special case. Some arguments provide stronger support for their conclusion than other arguments. For example, temporal projection provides weaker support for its conclusion as the time interval between t_0 and t_1 grows. Similarly, when a conclusion is drawn (defeasibly) on the basis of high probability, the higher the probability the better the justification for the conclusion. Degrees of justification should play a role in the computation of defeat statuses. Given a strong argument for P and a much weaker argument for $\sim P$, P should be undefeated. But if the arguments are of equal strength, both P and $\sim P$ should be defeated. Ideally, a semantics for defeasible reasoning should tell us not just which conclusions are justified, but how justified they are. The proposal I made in [31] and [32] was to define *status-assignments* to be assignments of degrees of justification to the conclusions of arguments, where the assignments are required to be consistent with certain intuitively plausible principles regarding degrees of justification. Then a conclusion is ruled undefeated iff it is undefeated in every status assignment.

More precisely, we collect arguments into an *inference graph*, where the nodes represent the conclusions of arguments, *support-links* tie nodes to the nodes from which they are inferred, and *defeat-links* indicate defeat relations between nodes. The analysis is somewhat simpler if we construct the inference graph in such a way that when the same conclusion is supported by two or more arguments, it is represented by a separate node for each argument. So in an important sense, the nodes represent arguments rather than just representing their conclusions. A node α has a propositional content prop(α) (the proposition *supported by* α), the set def(α) of nodes that are defeaters for α, the set basis(α) of nodes from which α is inferred, and the reason-scheme employed in the inference.

To define status-assignments, we first define the notion of a partial-status assignment, which assigns status-assignments to some subset of an inference-graph G in accordance with the following rules:

σ is a *partial-status-assignment* for an inference-graph G iff for each $\alpha \in G$:

1. if α encodes a percept, $\sigma(\alpha)$ is the strength of the percept;
2. otherwise, if basis(α) = $\varnothing$, $\sigma(\alpha) = 0$;
3. otherwise, if for some $\delta \in$ def(α), either $\sigma(\delta) \geq$ reason-strength(α) or there is a $\beta \in$ basis(α) such that $\sigma(\delta) \geq \sigma(\beta)$, $\sigma(\alpha) = 0$;
4. otherwise, if σ assigns values to all members of def(α) and basis(α), $\sigma(\alpha)$ = the minimum of reason-strength(α) and the values of $\sigma(\beta)$ for $\beta \in$ basis(α).

σ is a *status-assignment* for an inference-graph G iff σ is a maximal partial-status-assignment for G (i.e.,

there is no partial-status-assignment σ* for G such that σ ⊂ σ*).

A node α is *undefeated relative to* an inference-graph G iff for every status-assignment σ for G, σ assigns a non-zero value to α.

Different status-assignments may assign different values to a node, but it turns out that if the node is undefeated, all status-assignments assign the same value. So we can define:

If a node α is defeated relative to an inference-graph G, its degree of justification relative to G is 0. If it is undefeated, its degree of justification relative to G is the unique value assigned to it by every status-assignment.

Recall that nodes represent arguments. If a proposition *P* is supported by more than one argument, then it will be supported by more than one node in the inference-graph. Accordingly, we can define:

A proposition *P* is justified to degree δ relative to an inference-graph G iff δ is the maximal γ such that there is an α for which prop(α) = *P* and α is justified to degree γ relative to G.

A fully implemented system of defeasible reasoning based on this semantics is available on my web site. It is this system that was used for implementing the principles discussed in sections three and four.

However, the use of this semantics in recent work on decision-theoretic planning convinced me that it gives the wrong result in some important cases. The difficulty is that on this semantics, defeat is an all-or-nothing matter. If a defeater is less justified than what it defeats, then it has no effect. This seems intuitively wrong. Defeaters should be able to "diminish" the degree of justification of a conclusion even when they are not able to defeat the conclusion outright. In my [41] I made an initial proposal for a semantics that could accommodate diminishers. Unlike semantics based on multiple models or multiple status assignments, the new semantics provides a fully recursive characterization of degrees of justification. The general obstacle to giving a recursive characterization is that there can be inference/defeat-paths (paths consisting of linked support-links and defeat-links) that go in a circle. Consider, for example, the simple case of "collective defeat" diagrammed in figure 1. A straightforward recursive computation would require us to know the value of ~*Q* in order to compute that for *Q*, and similarly to know that for *Q* before computing that for ~*Q*. The new semantics solves this problem as follows. Let us say that a defeater for a node of the inference-graph is *independent* of the node iff there is no inference/defeat-path from the node to the defeater. First, it is shown that very general assumptions allow us to prove that if a defeater is independent of the node for which it is a defeater, then the affect is to simply subtract the strength of the defeating argument from the strength of the supporting argument (with the proviso that the value does not go below zero). This was argued on the basis of fairly strong assumptions in my [41], but it is shown in my [46] that the assumptions can be weakened dramatically. This handles the case in which the recursive computation is not circular. When we do have circularity, the semantics removes the circularity by deleting defeat-links that lead to the circularity. These are the *critical links*. In the above example, this produces the inference-graph of figure 2. The values of *Q* and ~*Q* are then computed recursively in this modified inference-graph. The degree of justification of *Q* in the original inference-graph is then the degree of justification of *Q* in the second inference-graph minus the degree of justification of ~*Q* in the second inference-graph.

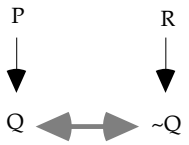


Figure 1. Collective Defeat

Figure 2. Removing Critical Links

To turn this into a general semantics, we must say which defeat-links are critical. The proposal of my [41] was that a defeat-link is critical iff it is a member of a minimal set of defeat-links whose deletion is sufficient to make *Q* not self-dependent. Last Spring, I produced an implementation of that semantics, but experimenting with the implementation led me to the realization that there were still some problems the semantics did not handle correctly. I now have a tentative fix, but it is not as elegant as the original theory, and that bothers me a bit. I am exploring three different approaches to implementing the semantics. Any of these approaches should work, but if the resulting system is to provide the inference engine for a cognitive agent it is important that it be as efficient as possible. Thus it is important to pursue all three approaches to implementation and compare the results experimentally. Given the implemented system, it will be possible to use it to look for any further difficulties for the semantics.

Two of the three approaches to implementation do not look much like the original semantics, and proving them correct involves proving some complex theorems in graph theory. I anticipate spending another year completing the theory and the implementation.

When the work is finished, it will constitute a theory of defeasible reasoning and an implemented AI system that can provide the inference engine for a general-purpose cognitive agent. My plan is to then return briefly to the reasoning described informally in section three. The work on perceptual reasoning, temporal projection, and causal reasoning will, I think, require only minor polishing, and it will in turn provide a test bed for the defeasible reasoner.

At the same time this work is being done, I will have graduate students working on improving information storage in the discrimination net. We can then test OSCAR's ability to reason defeasibly against the background of a huge database of information. It will be necessary for an autonomous agent to be able to do this. Working this out may take another six months.

6. Reasoning About Probabilities

For the past eight years, I have been pursuing research on decision-theoretic planning in autonomous agents, funded by NSF grants no. IRI-9634106 and IIS-0080888. This has produced a general theory of decision-theoretic planning that differs considerably from more familiar theories (see [39]-[45]). The work described in section three, and the theoretical work described in section five, were supported by this grant. The main product of the grant is a book, *Thinking about Acting: Logical Foundations for Rational Decision Making*, which is very close to finished and will probably be published by Oxford University Press. The intent is to use this theory for the basis of an implemented decision-theoretic planner, to be incorporated into OSCAR. However, before the theory can be implemented, I must have an implementation of certain kinds of probabilistic reasoning. After all, decision-theoretic reasoning makes heavy use of probabilities. Preliminary work on implementing this probabilistic reasoning is what revealed the above noted inadequacies concerning the way my theory of defeasible reasoning dealt with diminishers. I realized that to make further progress I had to solve the problems concerning the reasoner, so my work on planning was temporarily set aside. When the work described in section five is finished, I will return to the investigation of probabilistic reasoning, and ultimately to decision-theoretic planning. However, the construction of an implemented planner will probably not begin until after the period of this proposed grant. What I want to work on during the last half of the period of this grant is the probabilistic reasoning.

Probability Distributions Probabilities are important for both epistemic and practical cognition. The rover requires knowledge of probabilities both to make judgments about its current surroundings and to engage in decision-theoretic planning about what it should do given those current surroundings. Most work on decision-theoretic planning has pretended that the planning agent has at its disposal a complete probability distribution over all the relevant variables of the problem (for example, [2],[3],[5],[11],[15],[16],[17],[54],[55]). In other words, where $P_1,...,P_n$ is any set of propositions relevant to the problem (this includes propositions and their negations), the agent knows the value of **PROB**$(P_1\&...\&P_n)$. This probability distribution should reflect both initial knowledge built into the agent by the system designer and any new knowledge the agent has acquired by subsequent experience of its environment.

Everyone knows that in most real-world contexts, this is a completely unrealistic assumption, but I don't think it has generally been appreciated how serious the problem is, and researchers have not dealt with the problem of how to do probabilistic reasoning without having a complete probability assignment. First, consider the magnitude of the problem. For a sophisticated agent operating in a novel environment, pretty much anything *might* be relevant. It is up to the agent to discover what is relevant by discovering what variables affect the probabilities of outcomes. So there cannot be much a priori restriction on what variables are included in the probability distribution. But then it follows that it is impossible for a sophisticated agent operating in an environment of real-world complexity to have such a probability distribution at its disposal. The point is not that the agent could not learn the relevant probabilities. It could not even store them all. The problem is a simple cardinality problem. How many variables must be included in this probability distribution? If we have sufficiently limited aspirations, we may only insist that our agent be able to function in a narrowly circumscribed environment. It is hard to imagine any realistic environment of practical interest with, collectively, fewer than 300 variables that are relevant to the probabilities of some of the outcomes. If we want our agent to be able to deal with novel environments of unrestricted complexity, then the number of variables may be larger by many orders of magnitude. But let's suppose there are just 300 two-valued variables relevant to the planning problem. Then the number of conjunctions that must be assigned probabilities by a complete probability distribution is 2^{300}. This is approximately 10^{90}. This is an immense number. It has been estimated that the number of elementary particles in the universe is 10^{78}. 2^{300} is twelve orders of magnitude larger, and this is from just 300 relevant variables. Clearly, no agent can store a complete probability distribution for such a problem in the form of an explicit assignment of probabilities to each conjunction.

Perhaps there is a more efficient way of storing the probability distribution. For example, could we use a Bayesian net (see [18]) with just 300 nodes? Bayesian nets are only helpful if the nodes are sparsely connected, i.e.,

if most of the probabilities recorded in the net are statistically independent of most of the nodes. How sparse does the net have to be? Well, even if only one in every trillion (10^{12}) probabilities had to be explicitly recorded in the Bayesian net, that would still leave 10^{78} links, i.e., as many links as there are elementary particles in the universe. There is no way to store such a Bayesian net in a real agent.

Can we realistically suppose that our connections are even sparser — so sparse that it is possible to store the Bayesian net in a real agent? Let's consider an example. This generalizes Kushmerick, Hanks and Weld's [11] "slippery gripper" problem. We are presented with a table on which there are 300 numbered blocks, and a panel of numbered buttons. Pushing a button activates a robot arm which attempts to pick up the corresponding block and remove it from the table. We get 100 dollars for each block that is removed. Pushing a button costs two dollars. The hitch is that half of the blocks are greasy, but we don't know which. Initially each block has an equal probability of being greasy. If a block is not greasy, pushing the button will result in its being removed from the table with probability 1.0, but if it is greasy the probability is only 0.1. We are given 300 chances to either push a button or do nothing. In between, we are given the opportunity to look at the table, which costs one dollar, or do nothing. Looking will reveal what blocks are still on the table, but will not reveal directly whether a block is greasy. What should we do? Humans find this problem terribly easy. Everyone I have tried this upon has immediately produced the optimal plan: push each button once, and don't bother to look at the table. Although humans find this problem easy, I have argued recently [42] that no existing planner can solve this problem. But what is important for present purposes is that the probabilities of relevance to this problem cannot be represented by a Bayesian net. The relevant variables are whether a block is on the table (T_i), whether a button has been pushed (P_i), and whether a block is greasy (G_i). It would be natural to include a node for each of these (for each stage of the plan), producing a graph with 5200 nodes, and then linking the nodes as necessary to record the primitive probabilistic connections. However, the resulting graph would not be a Bayesian net, because a Bayesian net must be acyclic. If we include nodes for the greasiness of the blocks, acyclicity fails. The probability of a block being on the table after the corresponding button is pushed is influenced by whether it is greasy, and the probability of its being greasy given that the button is pushed is influenced by whether it is still on the table. If we do not include nodes for the greasiness of the blocks, then the nodes just concern which blocks are on the table at each stage and which buttons have been pushed. This more restricted set of nodes fails to represent all of our probabilistic information, but it does have the form of a Bayesian net. However, as noted above, the probability of a block being greasy is influenced by what other blocks are on the table, and that in turn affects the probability that the block will still be on the table after its button is pushed. Thus the Bayesian net must encode as primitive every probability of the form $\mathbf{PROB}(T_i/P_i \,\&\, \Pi_{j\in K}T_j)$ where K is a set of block numbers and $i\notin K$. There are 2^{300} such probabilities, so this Bayesian net cannot be encoded in a real agent. The upshot is that even in rather simple domains of real-world complexity, the use of Bayesian nets may not solve the problem of encoding a complete probability distribution in an agent.

It is to be emphasized that this example is unrealistic only in that, in real-world applications like planetary exploration, the number of variables potentially relevant to the probabilities of various outcomes will be orders of magnitude greater than 300. The upshot is that, even by using Bayesian nets and making reasonable assumptions about probabilistic independence, an agent cannot store a complete probability distribution, or even complete probability distributions for relatively small subsets of variables. It must make do with much spottier knowledge of probabilities, and try to acquire new probability knowledge as it needs it. How is this possible?

Thin Knowledge of Probabilities It seems to be unavoidable that agents capable of sophisticated cognition about the real world must learn their probabilities as they go, and they will never have anything approaching a complete probability distribution. We can put this by saying that they will have "thin" knowledge of probabilities. This is worse than just gappy knowledge. What they don't know will, of necessity, be orders of magnitude greater than what they do know. It is worth noting that this does not set probabilities apart from anything else. Real agents will have thin knowledge of just about everything. The world is just too big for an agent to know a significant proportion of all the facts about it. How can agents function with such thin knowledge? This is a problem of supreme importance if we want to design sophisticated agents capable of functioning in the real world. In many cases it is defeasible reasoning that enables human beings to bridge the gaps, allowing them to draw conclusions on the basis of the limited knowledge they have, and allowing them to assume that if they knew more, that would not upset those conclusions. For example, when we reason inductively we extrapolate from our observations and infer defeasibly that things we have not observed will have the same general properties as things we have observed. Similarly, perceptual reasoning builds in the defeasible assumption that things are they way they appear to be, and temporal

reasoning builds in a defeasible assumption that the properties of things do not change very fast (temporal projection).

When we turn to probabilities, we find that agents must also be able to function on the basis of thin knowledge. This gives rise to two questions. First, how can agents (e.g., human beings) acquire the limited probability knowledge they do have? Second, how can they get by with so little probability knowledge? I will argue that defeasible reasoning plays a crucial role in the answer to both of these questions.

Two Kinds of Probabilities In addressing questions about how to use probabilities in cognition, it is important to realize that there are major disputes about the foundations of probability theory. These disputes bear upon the properties of probabilities and how they can be used for cognitive tasks like planning. Furthermore, there is more than one kind of probability, and the different kinds of probabilities have somewhat different logical and mathematical properties. AI researchers are often well versed in "standard" mathematical probability theory, but not in the philosophical foundations. This can be a major problem, because standard mathematical probability theory is just a version of measure theory, and it is often not clear to what extent its results are applicable to "real" probabilities. For instance, mathematical probability theory assumes that probabilities are countably additive, but that assumption is at least debatable when applied to various kinds of "real" probability. Countable additivity has been rejected by most of the important writers in the foundations of probability theory, including de Finetti [4], Reichenbach [50], Jeffrey [10], Skyrms [53], Savage [52], and Kyburg [12]. Philosophical issues in the foundations of probability have direct relevance to how we can build agents that are able to reason probabilistically.

The most important division in the foundations of probability concern the difference between subjective and objective probabilities. Objective probabilities are supposed to represent objective facts about the way the world is. Subjective probabilities are reports of the degree of belief of a cognizer rather than factual statements about the environment [11,18,51,52]. I believe that subjective probability theory is subject to overwhelming difficulties, and does not provide an adequate foundation for the probabilistic reasoning that an agent must perform to get around in the world. For a detailed discussion of some of these difficulties, see chapter four of [47] and chapter three of [32]. The simplest reason for rejecting the use of subjective probabilities in agent design is that if, as subjectivists often propose, the only constraint on a rational agent's subjective probabilities is that they be coherent (conform to the probability calculus), then an agent can attach absolutely any probability to any contingent proposition as long as the probabilities associated with other propositions are adjusted so that the entire set of probabilities is coherent. The probabilities will be completely insensitive to the way the world is. But if we are going to use probabilities as a guide to action, surely we want them to reflect the way the world is. I admit that this may be an overly quick rejection of subjective probability theory, but see the above references for a more careful discussion of the matter.

Most theories of objective probability take there to be an intimate connection between probabilities and frequencies. See chapter one of [28] for a survey of objective probability theories. Relative frequencies relate properties. **freq**[A/B] is the proportion of B's that are A's. For example, we can talk about the frequency with which patches of sand of a certain color are soft. Where $\#A$ is the cardinality of the set of all A's, **freq**[A/B] = $\#(A\&B)/\#B$. The exact connection between objective probabilities and frequencies is controversial (my own theory is presented in [28]), but at the very least, there is an epistemological connection between them. Observing that the relative frequency of A's in a sample of B's is some number r gives us a defeasible reason for thinking that **prob**(A/B), the probability of an arbitrary B being an A, is approximately r. This inference is based upon a general principle of *statistical induction*, and one of the burdens of a theory of objective probability is to formulate such a principle precisely.

Objective probabilities inferred from relative frequencies have the same logical form as the relative frequencies themselves. That is, they relate properties. **prob**(A/B) is the probability of an arbitrary B being an A. This is not the probability of a proposition being true. This is an *indefinite probability*, or a "general" probability. Indefinite probabilities are most naturally formulated using free variables. For example, we might write "the probability of a patch of sand of this color being soft" as **prob**(x is soft/x is sand of this color).

Inductive reasoning and statistical sampling justify beliefs about indefinite probabilities, but the probabilities needed for decision making are the probabilities that particular propositions are true. For example, our rover might conclude inductively that the probability of sand being soft when it looks a certain way is .7. This is an indefinite probability about arbitrary times and places. But in deciding whether to attempt to cross this particular patch of sand, what the rover wants to know is how probable it is that *this very patch of sand* is soft. This is the probability of a particular proposition being true, viz., the proposition that this sand is soft. Such probabilities are *definite* or "single case" probabilities. I will follow the convention of symbolizing indefinite probabilities using **prob** and definite probabilities using **PROB**. Intuitively, both definite and indefinite probabilities make sense. An objective probability

theory must accommodate both. Introspecting our own cognition, it seems pretty clear that statistical or inductive reasoning produces knowledge of indefinite probabilities, and then definite probabilities are inferred by somehow applying the indefinite probabilities to particular cases. This kind of inference is called *direct inference*. I will discuss direct inference a bit more fully below.

It is noteworthy that standard mathematical probability theory is only a theory of definite probabilities, not indefinite probabilities. The basis for mathematical probability theory is Kolmogorov's axioms, and according to those axioms probabilities attach to "events", which are best identified with classes of logically equivalent propositions. Indefinite probabilities, dealing as they do with relations (expressed by open formulas in, e.g., first-order logic) have a richer logical structure than definite probabilities. There are numerous principles that hold for indefinite probabilities but cannot even be expressed in the language of the standard probability calculus. Here are three intuitively plausible ones that were discussed in [28]:

(IND) $\mathbf{prob}(Axy/Bxy\ \&\ y = c) = \mathbf{prob}(Axc/Bxc)$.

(PFREQ) $\mathbf{prob}(Ax/Bx\ \&\ \mathbf{freq}[Ay/By] = r) = r$.

(PPROB) $\mathbf{prob}(Ax/Bx\ \&\ \mathbf{prob}(Ay/By) = r) = r$.

None of these principles is even well-formed in the standard probability calculus. This is another reflection of the fact that mathematical probability theory may not have much to do with "real" probabilities.

Note that the free variables occurring in definite probabilities are quite different from the "random variables" occurring in the standard probability calculus. If r is a random variable ranging over patches of sand on Mars, then $\mathbf{PROB}(r$ is soft) is the probability distribution possibly assigning a different value to the definite probability of each patch of sand being soft. $\mathbf{PROB}(r$ is soft) does not have a single value. On the other hand, $\mathbf{prob}(x$ is soft/x is a patch of sand on Mars) has a single value. It is, roughly, the proportion of patches of sand on Mars that we would expect to be soft.

It is remarkable how often definite and indefinite probabilities are confused with one another in AI. For example, imagine a medical diagnosis system based on a Bayesian net. Clearly, the probabilities that go into building the net are general probabilities, i.e., indefinite probabilities. But the conclusions of medical diagnosis are the probabilities that specific patients have particular diseases, i.e., they are definite probabilities. Definite probabilities cannot be derived from indefinite probabilities just on the basis of calculations in the probability calculus. Direct inference is required, and as we will see below, that involves more than mathematical calculation. But all Bayesian nets can do is perform calculations in the probability calculus. So this use of Bayesian nets is mathematically invalid.

A sophisticated autonomous rover is going to have to be able to discover indefinite probabilities describing its environment (e.g., when the surface of the ground looks a certain way it is apt to be soft), employ direct inference to infer definite probabilities about its current situation (e.g., the sand in front of it now is probably soft), and then use the latter in decision-theoretic reasoning about what to do. To implement such reasoning in an agent, we first need precise theories about how statistical induction and direct inference should work. These are epistemological theories governing how to reason about indefinite and definite probabilities.

Direct Inference For decision-theoretic reasoning, an agent must know the probabilities of various possible outcomes of performing a specific action here and now. These are definite probabilities — not indefinite probabilities. For example, if the rover is faced with a patch of reddish sand, it will want to know the probability that if it attempts to drive over this patch of sand, it will become bogged down. What is at issue here is the definite probability. The rover should only be interested in the indefinite probability of becoming bogged down while driving over an arbitrary reddish patch of sand insofar as that helps it to evaluate the definite probability. The indefinite probability is of only "theoretical" interest. The definite probability is of pressing practical concern.

Although our practical interest is in the definite probabilities, it seems clear that the way we get them is by applying our knowledge of indefinite probabilities to the present circumstances. If the sand has a particular reddish cast, and I know that the probability is .95 of sand of that color being soft, then as long as I don't know anything special about this particular patch of sand that would affect the probability, I will infer that the probability that this patch of sand is soft is .95. In other words, I infer $\mathbf{PROB}(St) = .95$ from the facts that (1) $\mathbf{prob}(Sx/Rx) = .95$ and (2) Rt. This illustrates that although definite probabilities and indefinite probabilities are different beasts, we get definite probabilities by applying indefinite probabilities to particular situations. A theory of direct inference must explain how this works. Unfortunately, in many cases it is more difficult to see what inference to make. For example, I may know that $\mathbf{prob}(Sx/Rx) = .95$ and Rt, but also that $\mathbf{prob}(Sx/Ux) = .75$ and Ut. Then what should I conclude about

PROB(St)?

The basic idea behind direct inference was first articulated by Hans Reichenbach [50]: in determining the probability that an individual c has a property F, we find the narrowest reference class X for which we have reliable statistics and then infer that **PROB**(Fc) = **prob**($Fx/x{\in}X$). For example, insurance rates are calculated in this way. There is almost universal agreement that direct inference is based upon some such principle as this, although there is little agreement about the precise form the theory should take. There are, as far as I know, just four theories of direct inference that have been worked out in detail: Kyburg [12], Levi [13], Bacchus (1990), and Pollock [28]. Halpern (1990) discusses the distinction between definite and indefinite probabilities, but does not explore the topic of direct inference.

Direct inference proceeds by using what we know or are justified in believing about particular objects in particular situations to instantiate indefinite probabilities. If we are justified in believing G_1c, G_2c, and G_3c and we know that **prob**(Fx/G_1x & G_2x & G_3x) = r, this gives us a reason for believing that **PROB**(Fc) = r. However, such reasoning is subject to a "total evidence" requirement. If we are also justified in believing G_4c, and we know that **prob**(Fx/G_1x & G_2x & G_3x & G_4x) = $s \neq r$, then we should infer that **PROB**(Fc) = s. Thus the original inference must be defeasible, and it is defeated by acquiring additional justified beliefs that instantiate different indefinite probabilities.

As a first approximation, we can capture the dynamics of this reasoning using two principles:

(DI) "Gc & **prob**(Fx/Gx) = r" is a defeasible reason for "**PROB**(Fc) = r".

(SDI) "Hc & **prob**(Fx/Gx & Hx) ≠ **prob**(Fx/Gx)" is a defeater for (DI).

In the preceding example, by (DI) we have a defeasible reason for believing that **PROB**(Fc) = r, and also a defeasible reason for believing that **PROB**(Fc) = s. In the absence of any other defeaters, these two inferences (to incompatible conclusions) would defeat each other "collectively". However, by (SDI), we also have a defeater for the inference to the conclusion that **PROB**(Fc) = r, so that inference is defeated leaving the inference to the conclusion that **PROB**(Fc) = s undefeated. Thus we get the effect of the total evidence requirement (see my [37]).

Consider a concrete example. Suppose once more that the rover is deciding whether to attempt to drive over a patch of sand, and it wants to know the probability that it is soft. The sand has a particular reddish cast, and the rover knows that the probability of sand of that color being soft is .95. This gives it a defeasible reason for thinking that the probability is .95 that this sand is soft. However, the sand also has a certain texture, and the rover knows that the probability of sand being soft when it has that combination of color and texture is only .6. Then the rover also has a defeasible reason for thinking that the probability is .6 of this sand being soft. This conflict is resolved by (SDI), according to which the latter inference takes precedence over the former because it is based on more information.

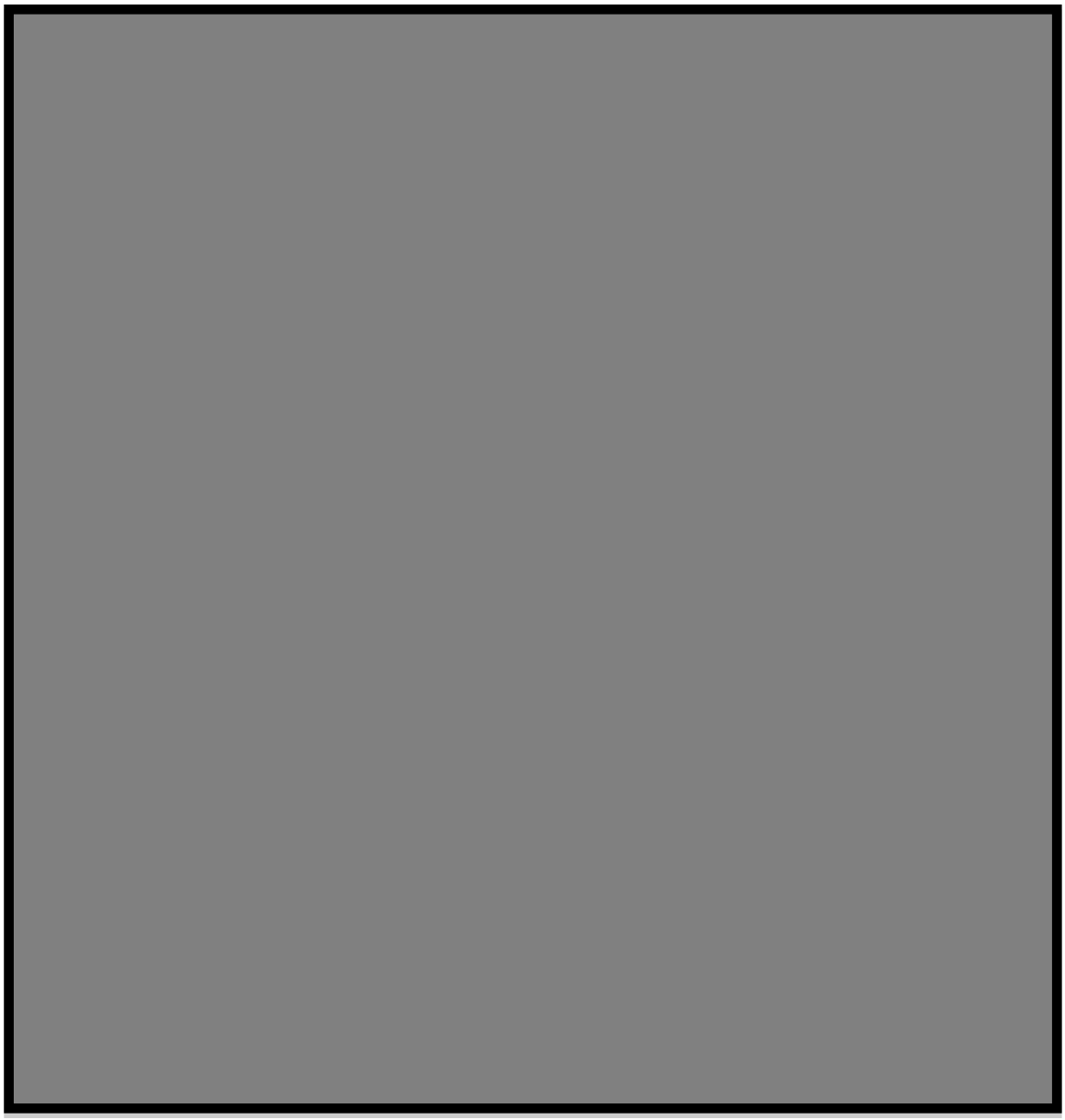
This is only a very rough sketch of a theory of direct inference. As Kyburg [12] was the first to note, more defeaters than just (SDI) are required to make the theory work. I proposed a more extensive theory in [28], but that theory is still not complete. The point I want to make here is just that the design of a sophisticated autonomous agent requires us to work out the details of some such theory of direct inference, and implement it. The theory sketched here makes essential use of defeasible reasoning, and I think this will be equally true of any adequate theory. So the theory must be implemented on top of a general-purpose defeasible reasoner. The completion of the theory of direct inference and its implementation in OSCAR will be my principal objective once the work described in section five is completed. It will occupy the last year and a half of the grant.

The purpose of this research is not just to work out a theory, but also to implement it within OSCAR. An implementation of this reasoning is necessary for implementing decision-theoretic planning in an agent operating in an environment of sufficient complexity that it cannot simply be given a complete probability distribution ahead of time by its designers.

7. Research Plan

The problem of building intelligent robots like a sophisticated autonomous rover is not going to be solved by engineering alone. Engineering needs sound theory, and in many cases the theory required for solving this problem still needs to be developed. OSCAR is closer to providing the necessary foundation than any other existing system, but there is still much work to be done. During the first 12 months, I will work on refining the new semantics for defeasible reasoning and developing an efficient implementation of it. During the next six months we will develop the database tools required to make OSCAR efficient when reasoning with a huge database of information. This will be based on the work sketched above. Then during the last 18 months I will turn to probabilistic reasoning. There is both theoretical work and implementational work to be done here. The basic theory is that of my [28], but in the

thirteen years since that book was published, some difficulties have been uncovered, and the theory must be refined to meet them. The theory must be implemented in a efficient manner. That will provide the ability to test the theory by applying it to realistic reasoning problems. This will also provide the inference engine for future work on decision-theoretic planning.



References Cited

[1] Bondarenko, A., Phan M. Dung, Robert Kowalski, and Francesca Toni, "An abstract, argumentation-theoretic approach to default reasoning", *AIJ* **93** (1997), 63-101.

[2] Boutelier, Craig, Thomas Dean, and Steve Hanks, "Decision theoretic planning: structural assumptions and computational leverage", *Journal of AI Research* **11** (1999), 1-94.

[3] Blythe, Jim, and Manuela Veloso, "Analogical replay for efficient conditional planning", *AAAI97.*

[4] de Finetti, B., *Theory of Probability*, vol. 1. New York: John Wiley and Sons, 1974.

[5] Draper, Denise, Steve Hanks, and Daniel Weld 1994 "Probabilistic planning with information gathering and contingent execution", *Proceedings of AIPS94.*

[6] Dung, P. M., "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *n*-person games", *Artificial Intelligence* **77** (1995), 321-357.

[7] Hanks, Steve, and Drew McDermott, "Default reasoning, nonmonotonic logics, and the frame problem", AAAI-86.

[8] Hanks, Steve, and Drew McDermott, "Nonmonotonic logic and temporal projection", *Artificial Intelligence* **33** (1987), 379-412.

[9] Israel, David, "What's wrong with non-monotonic logic?" *Proceedings of the First Annual National Conference on Artificial Intelligence*, 1980, 99–101.

[10] Jeffrey, Richard, *The Logic of Decision, 2nd edition*, University of Chicago Press, 1983.

[11] Kushmerick, N., Hanks, S., and Weld, D., "An algorithm for probabilistic planning". *Artificial Intelligence* **76** (1995), 239-286.

[12] Kyburg, Henry, Jr., *The Logical Foundations of Statistical Inference*. Dordrecht: Reidel, 1974.

[13] Levi, Isaac, *The Enterprise of Knowledge*. Cambridge, Mass.: MIT Press, 1980.

[14] Lin, Fangzhen, and Reiter, Raymond. "How to progress a database (and why) I. Logical foundations." In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation (KR'94).* 425-436.

[15] Majercik and Littman, "MAXPLAN: A new approach to probabilistic planning", *AIPS98*, 86-93.

[16] Majercik and Littman, "Contingent planning under uncertainty via stochastic satisfiability", *AAAI99.*

[17] Onder, Niluger, Martha Pollack, and John Horty, "A unifying algorithm for conditional, probabilistic planning", *AIPS1998 Workshop on Integrating Planning, Scheduling, and Execution in Dynamic and Uncertain Environments.*

[18] Pearl, Judea, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann, 1988.

[19] Pelletier, F. J., "Seventy-five problems for testing automatic theorem provers", *Journal of Automated Reasoning* **2** (1986), 191-216.

[20] Pollock, John, "Criteria and our Knowledge of the Material World", *The Philosophical Review*, **76** (1967), 28-60.

[21] Pollock, John, "What is an epistemological problem?", *American Philosophical Quarterly* **5** (1968), 183-90.

[22] Pollock, John, "The structure of epistemic justification", *American Philosophical Quarterly*, monograph series **4** (1970), 62-78.

[23] Pollock, John, "Perceptual Knowledge", *Philosophical Review*, **80** (1971), 287-319.

[24] Pollock, John, *Knowledge and Justification*, Princeton University Press, 1974.

[25] Pollock, John , "Epistemology and Probability", *Synthese* **81** (1983), 231-252.

[26] Pollock, John, *Contemporary Theories of Knowledge*, Rowman and Littlefield, 1986.

[27] Pollock, John, "Defeasible reasoning", *Cognitive Science* **11** (1987), 481-518.

[28] Pollock, John, *Nomic Probability and the Foundations of Induction*, Oxford University Press 1990.

[29] Pollock, John, "Interest driven suppositional reasoning", *Journal of Automated Reasoning* **6** (1990), 419-462.

[30] Pollock, John, "How to reason defeasibly", *Artificial Intelligence*, **57** (1992), 1-42.
[31] Pollock, John, "Justification and defeat", *Artificial Intelligence* **67** (1994), 377-408.
[32] Pollock, John, *Cognitive Carpentry*, MIT Press, 1995.
[33] Pollock, John, "Reason in a changing world", in *Practical Reasoning*, ed. Dov M. Gabbay and Hans Jürgen Ohlbach, Springer, 495-509, 1996.
[34] Pollock, John, "Implementing defeasible reasoning". Computational Dialectics Workshop at the International Conference on Formal and Applied Practical Reasoning, Bonn, Germany, 1996. This can be downloaded from http://www.u.arizona.edu/~pollock/.
[35] Pollock, John, "Taking perception seriously", in *Proceedings of the First International Conference on Autonomous Agents*, ACM Press, 1996.
[36] Pollock, John, "Reasoning about change and persistence: a solution to the frame problem", *Nous* **31** (1997), 143-169.
[37] Pollock, John, "Perceiving and reasoning about a changing world", *Computational Intelligence* **14** (1998), 498-562.
[38] Pollock, John, "Rational cognition in OSCAR", *Proceedings of ATAL-99*, ed. N. Jennings and Y. Lesperance, Springer Verlag, 2000.
[39] Pollock, John, "Locally global planning", *Proceedings of the Workshop on Decision-Theoretic Planning*, AIPS2000.
[40] Pollock, John, "Logical foundations for decision-theoretic planning", *Proceedings of the AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*, AAAI Press, 2001.
[41] Pollock, John, "Defeasible reasoning with variable degrees of justification", *Artificial Intelligence* **133** (2002), 233-282. A considerably expanded (and corrected) version of this paper is available at http://www.u.arizona.edu/~pollock.
[42] Pollock, John, "An easy 'hard problem' for decision-theoretic planners". Submitted to *Artificial Intelligence.* Available at http://www.u.arizona.edu/~pollock.
[43] "Against optimality: logical foundations for decision-theoretic planning in autonomous agents", submitted to *Decision Support Systems Journal* , for a special Issue on Decision Theory and Game Theory in Agent Design, available on my Web Site at http://www.u.arizona.edu/~pollock.
[44] Pollock, John, "Causal probability", *Synthese* **132** (2002), 143-185.
[45] Pollock, John, "Rational choice and action omnipotence", *Philosophical Review* **111** (2003) 1-23.
[46] Pollock, John, "Defeasible reasoning with variable degrees of justification II", unpublished expansion and revision of [53], available on my Web Site at http://www.u.arizona.edu/~pollock.
[47] Pollock, John, and Joseph Cruz, *Contemporary Theories of Knowledge*, 2nd edition, Lanham, Maryland: Rowman and Littlefield, 1999.
[48] Pollock, John, and Iris Oved, "Vision, knowledge, and the mystery link", forthcoming in *Epistemology: New Essays,* Quentin Smith (ed), Oxford University Press.
[49] Prakken, Henry and Gerard Vreeswijk, "Logics for Defeasible Argumentation", to appear in *Handbook of Philosophical Logic*, 2nd Edition, vol. 5, ed. D. Gabbay and F. Guenthner, Kluwer: Dordrecth, 2001.
[50] Reichenbach, Hans, *A Theory of Probability*. Berkeley: University of California Press, 1949. (Original German edition 1935)
[51] Reiter, Raymond, "A logic for default reasoning". *Artificial Intelligence* **13** (1980), 81–132.
[52] Savage, Leonard, *The Foundations of Statistics*, Dover, New York, 1954.
[53] Skyrms, Brian, *Causal Necessity*, Yale University Press, New Haven, 1980.
[54] Onder, Niluger, and Martha Pollack, "Contingency selection in plan generation", *ECP97*.
[55] Onder, Niluger, and Martha Pollack, "Conditional, probabilistic planning: a unifying algorithm and effective search control mechanisms", AAAI 99.
[56] Shanahan, Murray, *Solving the Frame Problem*, MIT Press, 1997.
[57] Vo, Q. B., N. Foo, and J. Thurbon, "Semantics for a theory of defeasible reasoning", submitted to *AMAI*.