Rational Cognition in OSCAR

John L. Pollock

Department of Philosophy
University of Arizona
Tucson, Arizona 85721
pollock@arizona.edu

1. Evaluating Agent Architectures

Stuart Russell [14] describes rational agents as "those that *do the right thing*". The problem of designing a rational agent then becomes the problem of figuring out what the right thing is. There are two approaches to the latter problem, depending upon the kind of agent we want to build. On the one hand, *anthropomorphic agents* are those that can help human beings rather directly in their intellectual endeavors. These endeavors consist of decision making and data processing. An agent that can help humans in these enterprises must make decisions and draw conclusions that are rational by human standards of rationality. Anthropomorphic agents can be contrasted with *goal-oriented agents* — those that can carry out certain narrowly-defined tasks in the world. Here the objective is to get the job done, and it makes little difference how the agent achieves its design goal.

If the design goal of a goal-oriented agent is sufficiently simple, it may be possible to construct a metric the measures how well an agent achieves it. Then the natural way of evaluating an agent architecture is in terms of the expected-value of that metric. An ideally rational goal-oriented agent would be one whose design maximizes that expected-value. The recent work on *bounded-optimality* ([3], [15], [20], etc.) derives from this approach to evaluating agent architectures. However, this approach will only be applicable in cases in which it is possible to construct a metric of success. If the design goal is sufficiently complex, that will be at least difficult, and perhaps impossible.

This paper will focus on anthropomorphic agents. For such agents, it is the individual decisions and conclusions of the agent that we want to be rational. In principle, we could regard an anthropomorphic agent as a special case of a goal-oriented agent, where now the goal is to make rational decisions and draw rational conclusions, but it is doubtful that we can produce a metric that measures the degree to which such an agent is successful in achieving these goals. It is important to realize that even if we could construct such a metric it would not provide an analysis of rationality for such an agent, because the metric would have to measure the degree to which the agent's individual cognitive acts tend to be rational. Thus it must presuppose prior standards of rationality governing individual cognitive acts.

In AI it is often supposed that the standards of rationality that apply to individual cognitive acts are straightforward and unproblematic, viz., Bayesian probability theory provides the standards of rationality for beliefs, and classical decision theory provides the standards of rationality for practical decisions. It may come as a surprise then that most philosophers reject Bayesian epistemology, and I believe there are compelling reasons for rejecting classical decision theory. Space precludes a detailed discussion of this issue, but I will say just a bit about why I think these theories should be rejected.

Bayesian epistemology asserts that the degree to which a rational agent is justified in believing something can be identified with a subjective probability. Belief updating is governed by conditionalization on new inputs. There is an immense literature on this. Some of the objections to it are summarized in Pollock and Cruz [11]. Perhaps the simplest objection to Bayesian epistemology is that it implies that an agent is always rational in believing any truth of logic, because any such truth has probability 1. This conclusion conflicts with common sense. Consider a complex tautology like $[P \leftrightarrow (Q \,\&\, {\sim}P)] \rightarrow {\sim}Q$. If one of my logic students picks this out of the air and believes it for no reason, we do not regard that as rational. He should only believe it if he has good reason to believe it. In other words, rational belief requires reasons, and that conflicts with Bayesian epistemology.

Classical decision theory has us choose acts one at a time on the basis of their expected values. What is wrong with this is that it is *courses of action* or *plans* that must be evaluated decision-theoretically, and individual acts become rational by being prescribed by rationally adopted plans. (See Pollock [8].) This suggests a minor modification to classical decision theory which applies it to plans rather than acts. On this view, a plan is adoptable just in case it has a higher expected value than any of its competitors. However, I will argue below that this does not work either. The evaluation of plans in terms of their expected values is more complicated that this. The difficulty arises from the fact that plans, unlike acts, are structured objects that can embed one another and aim at different (more or less comprehensive) sets of goals.

The design of an anthropomorphic agent requires a general theory of rational cognition. The agent's cognition must be rational by human standards. Cognition is a *process*, so this generates an essentially procedural concept of rationality. Many AI researchers have followed Herbert Simon [16] in rejecting such a procedural account, endorsing instead a satisficing account based on goal-satisfaction, but that is not applicable to anthropomorphic agents.

There is a problem, however, concerning how to understand procedural rationality. We do not necessarily want an anthropomorphic agent to model human cognition exactly. For example, there is a robust psychological literature [17] indicating that humans have to *learn* modus tollens. Modus ponens seems to be built into our cognitive architecture, but modus tollens is not. But surely there would be nothing wrong with building modus tollens into the inferential repertoire of an anthropomorphic agent. We want such an agent to draw rational conclusions and make rational decisions, but it need not do so in exactly the same way humans do it. How can we make sense of this? Stuart Russell [15] (following Herbert Simon) suggests that the appropriate concept of rationality should only apply to the *ultimate results* of cognition, and not the course of cognition. To make this more precise, let us say that a conclusion or decision is *warranted* (relative to a system of cognition) iff it is endorsed “in the limit”, i.e., there is some stage of cognition at which it becomes endorsed and beyond which the endorsement is never retracted. Then we might require an agent architecture to have the same theory of warrant as human rational cognition. This is to evaluate its behavior in the limit.

However, there is a problem for any assessment of agents in terms of the results of cognition in the limit. An agent that drew conclusions and made decisions at random for the first ten million years, and then started over again reasoning just like human beings would have the same theory of warrant, but it would not be a good agent design.

It looks like the most we can require is that the agent’s reasoning never strays very far from the course of human reasoning. If humans will draw a conclusion within a certain number of steps, the agent will do so within a “comparable” number of steps, and if a human will retract the conclusion within a certain number of further steps, the agent will do so within a “comparable” number of further steps. However, it has to be admitted that this is vague. A natural proposal for making this more precise might be to require that the worst-case difference be polynomial in the number of steps, but this seems pretty artificial. Furthermore, this particular proposal would not rule out the agent that drew conclusions randomly for the first ten million years.

I am not going to endorse a solution to this problem. I just want to call attention to it, and urge that whatever the solution is, it seems reasonable to think that the kind of architecture I am about to describe satisfies the requisite constraints.

2. The OSCAR Architecture

OSCAR is an architecture for rational agents based upon an evolving philosophical theory of rational cognition. The general architecture is described in Pollock [8], and related papers can be downloaded from http://www.u.arizona. edu/~pollock.

OSCAR is based on a schematic view of rational cognition according to which agents have beliefs representing their environment and an evaluative mechanism that evaluates the world as represented by their beliefs. They then engage in activity designed to make the world more to their liking. This is diagrammed in figure 1. This schematic view of rational cognition makes it natural to distinguish between *epistemic cognition*, which is cognition about what to believe, and *practical cognition*, which is cognition about what to do. We can think of the latter as including goal selection, plan construction, plan selection, and plan execution.

It is probably fair to say that most work on rational agents in AI has focussed on practical cognition rather than epistemic cognition, and for good reason. The whole point of an agent is *to do something*, to interact with the world, and such interaction is driven by practical cognition. From this perspective, epistemic cognition is subservient to practical cognition.

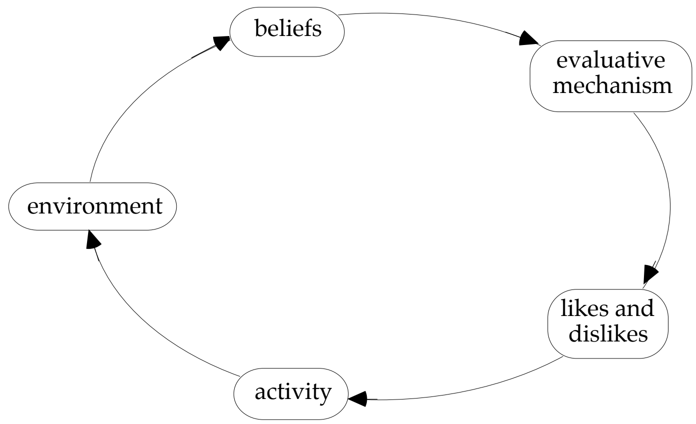

Figure 1. Doxastic-conative loop

The OSCAR architecture differs from most agent architectures in that, although it is still practical cognition that directs the agent's interaction with the world, most of the work in rational cognition is performed by epistemic cognition. Practical cognition evaluates the world (as represented by the agent's beliefs), and then poses queries concerning how to make it better. These queries are passed to epistemic cognition, which tries to answer them. Plans are produced by reasoning about the world (epistemic cognition). Competing plans are then evaluated and selected on the basis of their expected utilities, and those expected utilities are again computed by epistemic cognition. Finally, plan execution generally requires a certain amount of monitoring to verify that things are going as planned, and that monitoring is again carried out by epistemic cognition. In general, *choices* are made by practical cognition, but the information on which the choices are based is the product of epistemic cognition, and the bulk of the work in rational cognition goes into providing that information.

Epistemic cognition and practical cognition interact in important ways. The point of epistemic cognition is to provide the information required for practical cognition. This information is encoded in the form of beliefs, and the beliefs are used by all three modules of practical cognition, as diagrammed in figure 2.

The purpose of epistemic cognition is to provide the information used by practical cognition. As such, the course of epistemic cognition must be driven by practical cognition. It tries to answer questions posed in the pursuit of the solution of practical problems (the *ultimate epistemic interests*). Such reasoning is *interest-driven.* A certain amount of reasoning is also driven directly by the input of new information. So the basic interface between epistemic and practical cognition is as diagrammed in figure 3.

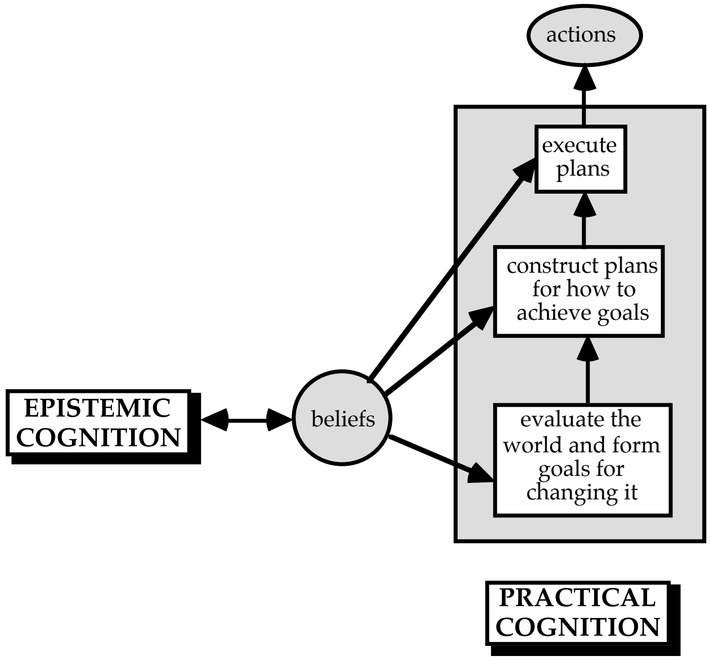

Figure 2. The subservient role of epistemic cognition

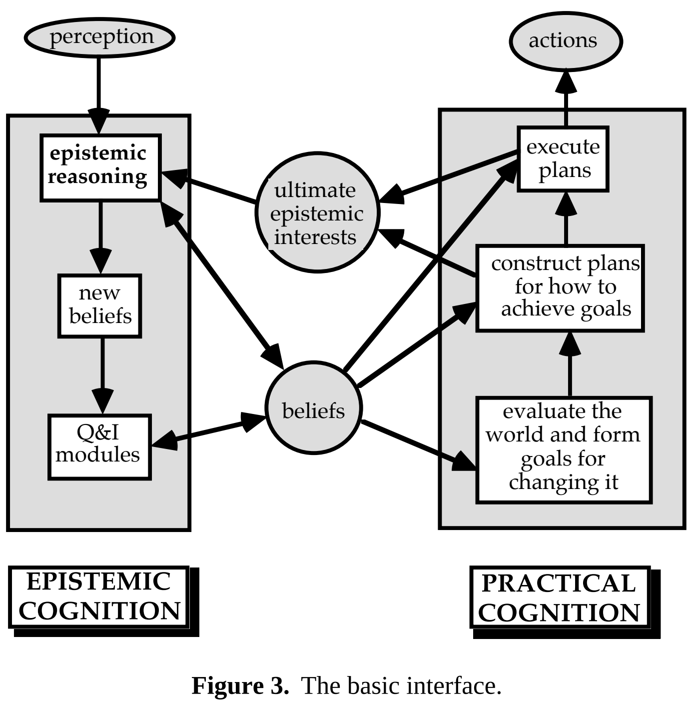

Figure 3. The basic interface.

Figure 3 also indicates a distinction between two kinds of epistemic cognition — epistemic reasoning, and Q&I modules. Although it is hard to say precisely what defines reasoning, we can say at least that it is serial, introspectible, and slow. A great deal of human cognition is performed instead by special-purpose Q&I (quick and inflexible) modules. For example, one could try to catch a baseball by reasoning about its trajectory, but that would be too slow. Instead, we employ a cognitive module whose sole purpose is the calculation of trajectories. This module achieves speed in part by making assumptions about the environment, e.g., that the ball will not bounce off of obstacles. There is reason to believe that humans have many built-in Q&I modules, with reasoning serving primarily to monitor and correct their output and to enable us to address questions for which we have no Q&I modules. This would seem to be a good architecture for agents in general. The Q&I modules provide speed, and reasoning provides generality.

It is often overlooked that it may not be possible to answer the questions posed by practical cognition simply by reasoning from previously held beliefs. Instead, the agent may have to "examine the world". E.g., it may have to discover what time it is by looking at a clock, or count the number of apples in a barrel, or look something up in an encyclopedia, or in an extreme case, send a spacecraft to Mars or run an experiment on a linear accelerator. Actions performed to acquire information constitute *empirical investigation*. Empirical investigation requires interacting with the world, and such interaction is driven by practical cognition, so the result is a loop wherein empirical investigation gives rise to *epistemic goals*, which are passed to practical cognition. Practical cognition then adopts interest in finding plans for achieving the epistemic goals, and passes that interest back to epistemic cognition. Thus epistemic and practical cognition are interleaved.

Plans for achieving epistemic goals often include actions controlling sensors. There is a distinction between active and passive perception, analogous to the distinction between looking and seeing. In the basic interface, perception is essentially passive, just feeding information to epistemic cognition. But perception can be controlled by, e.g., controlling the direction of the sensors, thus controlling what the agent is looking at. This constitutes "active perception". Empirical investigation can be combined with the basic interface to produce the diagram in figure 4.

Sophisticated agents can also engage in *reflexive cognition* — cognition about cognition. Applying practical cognition to reasoning can enable the agent to alter the course of its own reasoning, deciding what to think about and what strategies to use in problem solving. It is an open question how much power the agent should have over its own cognition. For example, it is reasonable for the agent to be able to alter the priority of cognitive tasks waiting to be performed, but presumably we do not want an agent to be able to make itself believe something just because it wants to. Adding the ability to redirect cognition, we get the general architecture diagrammed in figure 5.

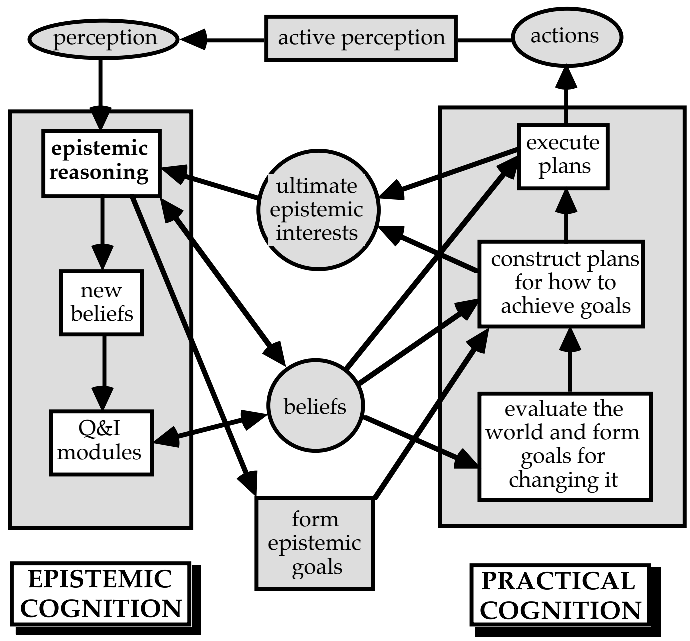

Figure 4. Empirical investigation

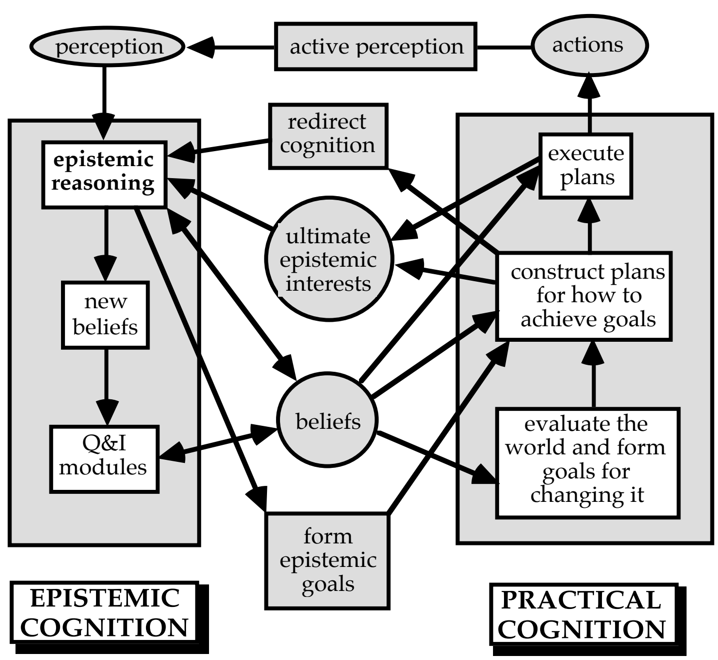

Figure 5. The Architecture for Rational Cognition

3. Epistemic Reasoning

Now let us look more closely at the structure of epistemic reasoning in OSCAR. We have seen that epistemic reasoning is driven by both input from perception and queries passed from practical cognition. This is accomplished in the OSCAR architecture by bidirectional reasoning. The queries passed to epistemic cognition from practical reasoning are *epistemic interests*, and OSCAR reasons backwards from them to other epistemic interests. In addition, OSCAR reasons forwards from beliefs to beliefs, until beliefs are produced that discharge the interests.

OSCAR is based upon a "natural deduction" theorem prover, rather than a more traditional resolution refutation theorem prover. This is because one of the principle objectives in building OSCAR was to produce a defeasible reasoner (see below). Resolution refutation makes essential use of reductio ad absurdum, and the latter is invalid for defeasible argumentation. Reasoning defeasibly to conflicting conclusions defeats the reasoning rather than disproving the premises.

Perhaps the most novel aspect of OSCAR's bidirectional reasoning is that reason-schemas are segregated into backward and forward schemas. Forward schemas lead from conclusions to conclusions. Backward schemas lead from interests to interests. For example, *simplification* (infer the conjuncts from a conjunction) is a fine rule to use in forward inference, but it would be combinatorially explosive when used in backward inference. Given an interest in *P*, it would have us adopt interest in every conjunction containing *P* as a conjunct. Similarly, *adjunction* (infer a conjunction from its conjuncts) is a natural rule to use for backward inference — given an interest in a conjunction, adopt interest in its conjuncts. But used in forwards inference it would have the reasoner form arbitrary conjunctions of its beliefs. A more sensible reasoner only forms conjunctions when they are of interest.

The motivation for building OSCAR in this way was to provide a platform for defeasible reasoning, but OSCAR turns out to be surprising efficient as a deductive reasoner. In a recent comparison with the highly respected OTTER resolution-refutation theorem prover (http://www.mcs.anl.gov/home/mccume/ar/otter) on a set of 163 problems chosen by Geoff Sutcliffe from the TPTP theorem proving library (Suttner and Sutcliffe, [18]), OTTER failed to get 16 while OSCAR failed to get 3. On problems solved by both theorem provers, OSCAR (written in LISP) was on the average 40 times faster than OTTER (written in C).

The principal virtue of OSCAR's epistemic reasoning is not that it is an efficient deductive reasoner, but that it is capable of performing defeasible reasoning. Deductive reasoning guarantees the truth of the conclusion given the truth of the premises. Defeasible reasoning makes it reasonable to accept the conclusion, but does not provide an irrevocable guarantee of its truth. Conclusions supported defeasibly might have to be withdrawn later in the face of new information. *All sophisticated epistemic cognizers must reason defeasibly*. This is illustrated by the following considerations:

- Perception is not always accurate. In order for a cognizer to correct for inaccurate perception, it must draw conclusions defeasibly and be prepared to withdraw them later in the face of conflicting information.

- A sophisticated cognizer must be able to discover general facts about its environment by making particular observations. Obviously, such inductive reasoning must be defeasible because subsequent observations may show the generalization to be false.
- Sophisticated cognizers must reason defeasibly about time, projecting conclusions drawn at one time forwards to future times (temporal projection). For example, consider a robot that can draw conclusions about its environment on the basis of perceptual input. It is given the task of comparing two meters, determining which has the higher reading. It examines the first meter and see what it reads, then turns to the second meter and reads it. But it is not yet in a position to compare them, because while examining the second meter it only knows what the first meter read *a moment ago*, not what it reads now. Humans solve this problem by assuming that the first meter has not changed in the short amount of time it took to read the second meter, but this is obviously a defeasible assumption because it *could have changed*. This illustrates that perception is really a form of sampling, telling a cognizer about small parts of the world at different times, and if the cognizer is to be able to put different perceptions together into a coherent picture of the world it must be allowed to assume defeasibly that the world does not change too fast.
- It will be argued below that certain aspects of planning must be done defeasibly in an autonomous agent operating in a complex environment.

Defeasible reasoning is performed using defeasible reason-schemas. What makes a reason-schema defeasible is that inferences in accordance with it can be defeated. OSCAR recognizes two kinds of defeaters. *Rebutting defeaters* attack the conclusion of the inference. *Undercutting defeaters* attack the connection between the premise and the conclusion. An undercutting defeater for an inference from P to Q is a reason for believing it false that P would not be true unless Q were true. This is symbolized $(P \otimes Q)$. More simply, $(P \otimes Q)$ can be read "P does not guarantee Q". For example, something's looking red gives us a defeasible reason for thinking it is red. A reason for thinking it isn't red is a rebutting defeater. However, if we know that it is illuminated by red lights, where that can make something look red when it isn't, that is also defeater but it is not a reason for thinking the object isn't red. Thus it constitutes an undercutting defeater.

Reasoning defeasibly has two parts, (1) constructing arguments for conclusions and (2) deciding what to believe given a set of interacting arguments some of which support defeaters for others. The latter is done by computing defeat statuses and degrees of justification given the set of arguments constructed. OSCAR uses the defeat-status computation described in Pollock [8].[1] This defeat status computation proceeds in terms of the agent's *inference-graph*, which is a data structure recording the set of arguments thus far constructed. We then define:

A *partial-status-assignment* for an inference-graph G is an assignment of "de-

---

[1] For comparison with other approaches, see [12].

feated" and "undefeated" to a subset of the arguments in *G* such that for each argument *A* in *G*:

1. if a defeating argument for an inference in *A* is assigned "undefeated", *A* is assigned "defeated";
2. if all defeating arguments for inferences in A *A* are assigned "defeated", *A* is assigned "undefeated".

A *status-assignment* for an inference-graph *G* is a maximal partial-status-assignment, i.e., a partial-status-assignment not properly contained in any other partial-status-assignment.

An argument *A* is *undefeated* relative to an inference-graph *G* of which it is a member if and only if every status-assignment for *G* assigns "undefeated" to *A.*

A belief is *justified* if and only if it is supported by an argument that is undefeated relative to the inference-graph that represents the agent's current epistemological state.

Justified beliefs are those undefeated given the current stage of argument construction. *Warranted conclusions* are those that are undefeated relative to the set of all possible arguments that can be constructed given the current inputs. Raymond Reiter [13] and David Israel [4] both observed in 1980 that when reasoning defeasibly in a rich logical theory like first-order logic, the set of warranted conclusions will not generally be recursively enumerable. This is because determining whether an argument is defeated requires detecting consistency, and by Church's theorem the latter is not recursively enumerable. This has the consequence that it is impossible to build an automated defeasible reasoner that produces all and only warranted conclusions. In other words, a defeasible reasoner cannot look much like a deductive reasoner. The most we can require is that the reasoner systematically modify its belief set so that it comes to approximate the set of warranted conclusions more and more closely. More precisely, the rules for reasoning should be such that:

(1) if a proposition *P* is warranted then the reasoner will eventually reach a stage where *P* is justified and stays justified;

(2) if a proposition *P* is unwarranted then the reasoner will eventually reach a stage where *P* is unjustified and stays unjustified.

It is shown in Pollock [8] that this is possible if the reason-schemas are "well behaved".

Given the ability to perform general-purpose defeasible reasoning, we can then provide an agent with reason-schemas for reasoning about specific subject matters. For example, OSCAR makes use of the following reason-schemas:

PERCEPTION

Having a percept at time *t* with content *P* is a defeasible reason to believe *P*-at-*t.*

PERCEPTUAL-RELIABILITY
“*R* is true and having a percept with content *P* is not a reliable indicator of *P*’s being true when *R* is true” is an undercutting defeater for PERCEPTION.

TEMPORAL-PROJECTION
“*P*-at-*t*” is a defeasible reason for “*P*-at-(t+Δt)”, the strength of the reason being a monotonic decreasing function of Δt.

STATISTICAL-SYLLOGISM
“*c* is a *B* & prob(*A*/*B*) is high” is a defeasible reason for “*c* is an *A*”.

To illustrate the use of these reason-schemas, consider the following problem. First, Fred looks red to me. Later, I am informed by Merrill that I am then wearing blue-tinted glasses. Later still, Fred looks blue to me. All along, I know that the probability is not high of Fred being blue given that Fred looks blue to me but I am wearing blue-tinted glasses. What should I conclude about the color of Fred? Intuitively, Fred’s looking red gives me a reason for thinking that Fred is red. Being informed by Merrill that I am wearing blue-tinted glasses gives me a reason for thinking I am wearing blue-tinted glasses. Fred’s later looking blue gives me a reason for thinking the world has changed and Fred has become blue. However, my knowledge about the blue-tinted glasses defeats the inference to the conclusion that Fred is blue, reinstating the conclusion that Fred is red. OSCAR’s reasoning is diagrammed by figure 6, where the “fuzzy” arrows indicate defeat relations.[2]

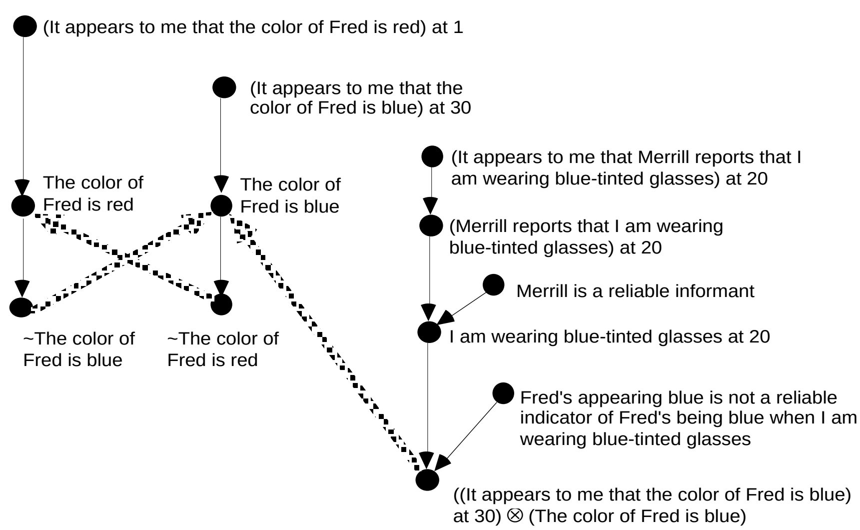

Figure 6. Inference graph

[2] For more details on this, see [9].

For a rational agent to be able to construct plans for making the environment more to its liking, it must be able to reason causally. In particular, it must be able to reason its way through the frame problem. OSCAR implements a solution to the frame problem. It has three constituents:

TEMPORAL-PROJECTION, discussed above.

CAUSAL-IMPLICATION, which allows us to make inferences on the basis of causal knowledge:

> If $t^* > t$, "A-at-t and P-at-t and (A when P is causally-sufficient for Q)" is a defeasible reason for "Q-at-t^*".

CAUSAL-UNDERCUTTER, which tells us that inferences based on causal knowledge take precedence over inferences based on temporal projection:

> If $t_0 < t < t^*$, "A-at-t and P-at-t and (A when P is causally-sufficient for ~Q)" is an undercutting defeater for the inference from Q-at-t_0 to Q-at-t by TEMPORAL-PROJECTION.

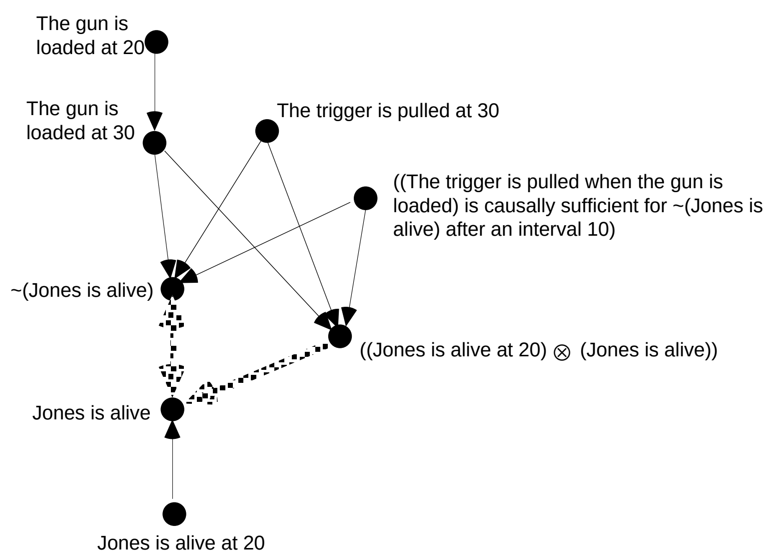

Figure 7. The Yale Shooting Problem

These principles can be illustrated by applying them to the Yale shooting problem [2]. I know that the gun being fired while loaded will cause Jones to become dead. I know that the gun is initially loaded, and Jones is initially alive. Later, the gun is fired. Should I conclude that Jones becomes dead? Yes, defeasibly. OSCAR solves this problem by reasoning as in figure 7. By TEMPORAL-PROJECTION, OSCAR has a reason to think that Jones will be alive. By CAUSAL-

IMPLICATION, OSCAR has a reason to think that Jones will be dead. By CAUSAL-UNDERCUTTER, the latter takes precedence, defeating the former.[3]

4. Practical Cognition

Given an agent capable of sophisticated epistemic cognition, how can it make use of that in practical cognition? We can regard practical cognition as having four components: goal selection, plan-construction, plan-selection, plan-execution. Although it is natural to think of these as components of practical cognition, most of the work will be carried out by epistemic cognition. To illustrate this, I will focus on plan-construction.

Standard planning algorithms assume that we come to the planning problem with all the knowledge needed to solve it. This assumption fails for autonomous rational agents. The more complex the environment, the more the agent will have to be self-sufficient for knowledge acquisition. The principal function of epistemic cognition is to provide the information needed for practical cognition. As such, the course of epistemic cognition is driven by practical interests. Rather than coming to the planning problem equipped with all the knowledge required for its solution, the planning problem itself directs epistemic cognition, focusing epistemic endeavors on the pursuit of information that will be helpful in solving current planning problems. Paramount among the information an agent may have to acquire in the course of planning is knowledge about the consequences of actions under various circumstances. Sometimes this knowledge can be acquired by reasoning from what is already known. But often it will require empirical investigation. Empirical investigation involves acting, and figuring out what actions to perform requires further planning. So planning drives epistemic investigation, which may in turn drive further planning. In autonomous rational agents operating in a complex environment, planning and epistemic investigation must be interleaved.

I assume that rational agents will engage in some form of goal-regression planning. This involves reasoning backwards from goals to subgoals whose achievement will enable an action to achieve a goal. Such reasoning proceeds in terms of causal knowledge of the form "performing action A under circumstances C is causally sufficient for achieving goal G". This is symbolized by the *planning-conditional* $(A/C) \Rightarrow G$.

A generally recognized problem for goal-regression planning is that subgoals are typically conjunctions. We usually lack causal knowledge pertaining directly to conjunctions, and must instead use causal knowledge pertaining to the individual conjuncts. We plan separately for the conjuncts of a conjunctive subgoal. When we merge the plans for the conjuncts, we must ensure that the separate plans do not destructively interfere with each other (we must "resolve threats"). Conventional planners assume that the planner already knows the consequences of actions under all circumstances, and so destructive interference can be detected by just checking

[3] For more details on this, see [9].

the consequences. However, an autonomous rational agent may have to engage in arbitrarily much epistemic investigation to detect destructive interference. Even if threats could be detected simply by first-order deductive reasoning, the set of threats would not be recursive. The following theorem is proven in Pollock [10]:

> If the set of threats is not recursive, then the set of planning ⟨problem,solution⟩ pairs is not recursively enumerable.
>
> Corollary: A planner that insists upon ruling out threats before merging plans for the conjuncts of a conjunctive goal may never terminate.

If the set of threats is not recursive, a planner must operate defeasibly, *assuming* that there are no threats unless it has reason to believe otherwise. That a plan will achieve a goal is a factual matter, of the sort normally addressed by epistemic cognition. So we can perform plan-search by adopting a set of defeasible reason-schemas for reasoning about plans. The following are examples of such reason-schemas:

GOAL-REGRESSION

> Given an interest in finding a plan for achieving G-at-t, adopt interest in finding a planning-conditional $(A/C) \Rightarrow G$. Given such a conditional, adopt interest in finding a plan for achieving C-at-t^*. If it is concluded that a plan *subplan* will achieve C-at-t^*, construct a plan by (1) adding a new step to the end of *subplan* where the new step prescribes the action A-at-t^*, (2) adding a constraint $(t^* < t)$ to the ordering-constraints of *subplan*, and (3) adjusting the causal-links appropriately. Infer defeasibly that the new plan will achieve G-at-t.

SPLIT-CONJUNCTIVE-GOAL

> Given an interest in finding a plan for achieving (G_1-at-t_1 & G_2-at-t_2), adopt interest in finding plans $plan_1$ for G_1-at-t_1 and $plan_2$ for G_2-at-t_2. Given such plans, infer defeasibly that the result of merging them will achieve (G_1-at-t_1 & G_2-at-t_2).

A number of additional reason-schemas are also required, but a complete planner can be constructed in this way. To illustrate, consider Pednault's [6] briefcase example. My briefcase and paycheck are initially at home, and the paycheck is in the briefcase. My goal is to have the briefcase at the office but the paycheck at home. OSCAR begins by producing the plan diagrammed in figure 8. This is a flawed plan, because taking the briefcase to the office also takes the paycheck to the office. Having produced this plan defeasibly, OSCAR undertakes a search for defeaters, and finds the defeating subplan of figure 9. OSCAR then fixes the flawed plan by adding a step that defeats the defeater, as in figure 10. The final plan is then that of figure 11.

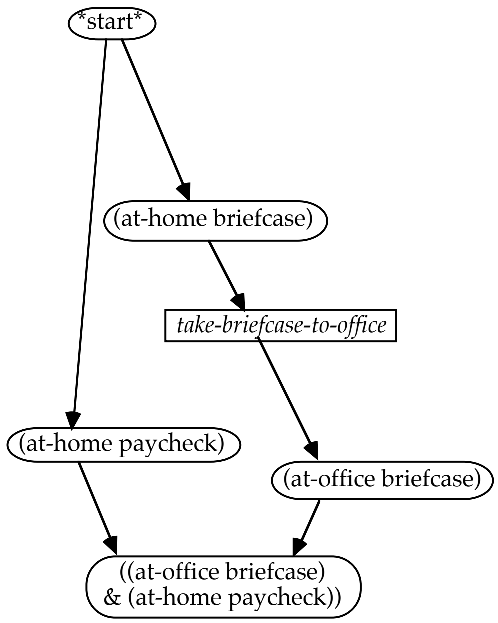

Figure 8. Flawed plan

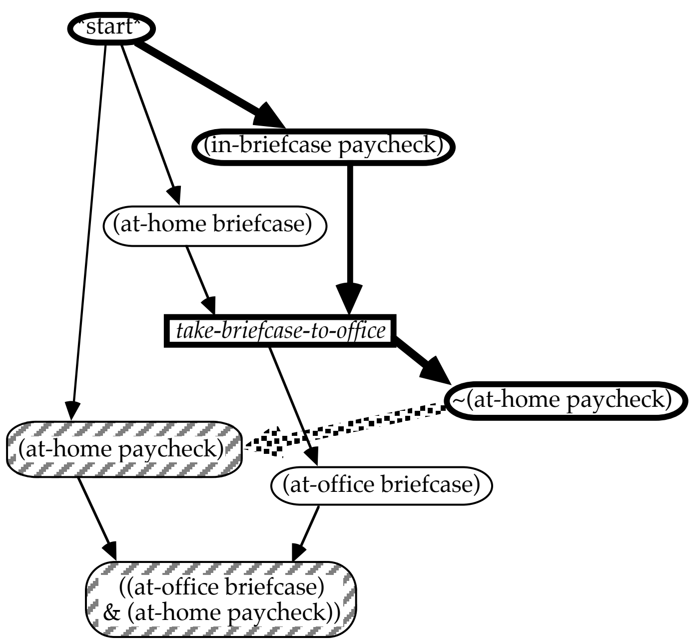

Figure 9. Defeating subplan

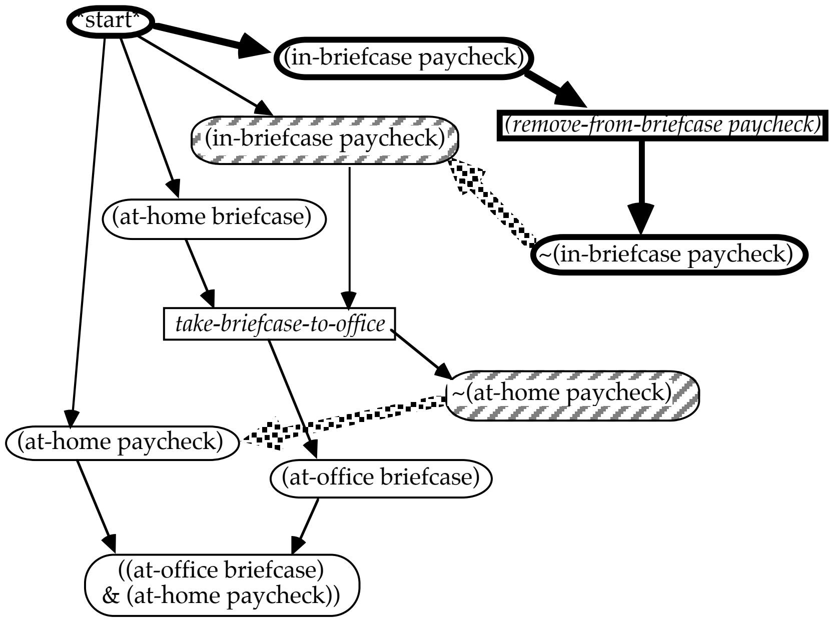

Figure 10. Defeating the defeater

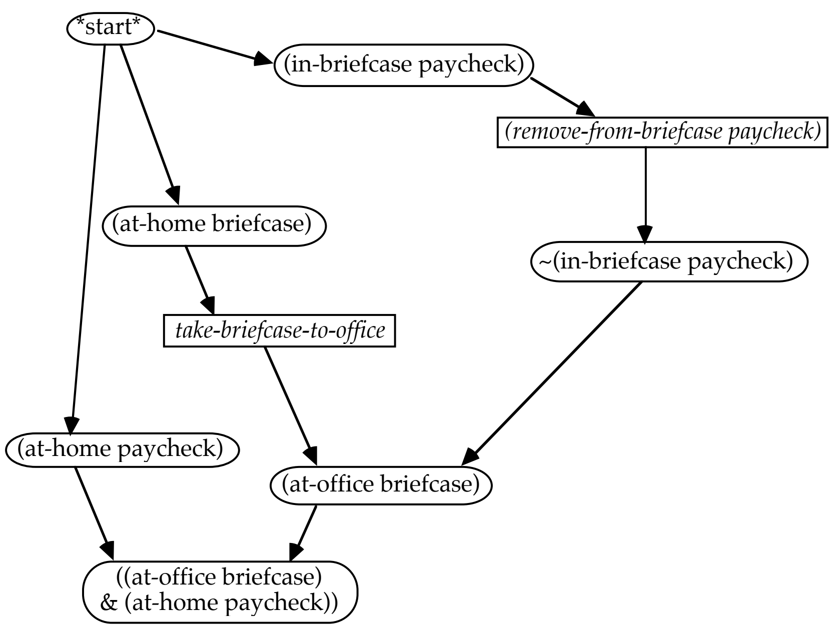

Figure 11. The final plan

Given a recursive list of consequences for each action, this planner will produce the same plans as conventional planners like UCPOP [7] or PRODIGY [19]. But when the list of consequences is nonrecursive the conventional planners will not return solutions, whereas OSCAR will still return the same solutions defeasibly. The search for defeaters may never terminate, but when the agent must act it can do so using its currently justified conclusions, which include the plans that were found defeasibly.

5. Adopting Plans

Thus far, everything that I have talked about has been implemented. Now I come to work in progress, and this is not implemented.

Plan construction produces plans that purport to achieve their goals, but *adopting* such a plan requires a further cognitive step. Such a plan is not automatically adoptable. For example, its execution costs might be greater than the value of the goal achieved. Or it might interact adversely with other plans already adopted, increasing their execution costs or lowering the value of their goals. Sometimes the impacted plan should be rejected, and sometimes the new plan should be rejected. In deciding whether to adopt a plan, we must evaluate it in a roughly decision-theoretic manner. This might suggest the use of Markov-decision planning (MDP's). However, a generally recognized problem for MDP's is computational infeasibility in complex domains. Although a lot of current research is directed at alleviating this problem [1], I am betting that there is no ultimate solution to it. So we must look for another way of doing decision-theoretic planning.

My proposal is that it is possible to perform feasible decision-theoretic planning by modifying conventional goal-regression planning in certain ways. Goal-regression planning can be performed by applying classical planning algorithms but appealing to probabilistic connections rather than exceptionless causal connections. This is computationally easier than standard "probabilistic planning" in the style, e.g., of BURIDAN [5], which uses probabilities to drive the planning. On my proposal, the planning is done conventionally and then probabilities computed later. In this connection, it is important to realize that it isn't really the probability of the plan achieving its goals that is important — it is the expected value. The expected value can be high even with a low probability if the goals are sufficiently valuable.

Once a plan is constructed, an expected value can be computed. This computation can be simplified by doing it defeasibly. An initial computation can be made using just the probabilities of plan-steps having their desired outcomes in isolation. Then a search can be undertaken for conditions established by other steps of the plan that alter the probabilities. This is analogous to the search for threats in deterministic causal-link planning.

Similarly, the initial computation uses default values for the execution costs of plan steps and the values of goals. However, these values can be different in the context of conditions established by other steps of the plan. Clearly, earlier steps can change the execution costs of later steps. For example, if a step transports a package from one point to another by truck, and an earlier step moves the truck, that can change the execution cost. Goals don't have fixed values either. For example, suppose my goal is to have a dish of vanilla ice cream, and a playful friend offers to give me one if I eat a dill pickle first. I may then construct the plan to eat a dill pickle in order to obtain the dish of ice cream. However, the value of the goal is greatly diminished by eating the pickle. So a search must be undertaken for conditions that alter execution costs and the values of goals. This is also analogous to the search for threats.

Obviously, expected values can be increased by adding conditional steps. Less obviously, expected values can be increased by planning hierarchically. A high-level plan can have a higher expected value than any of its low-level specifications. This is because we may be confident of being able to fix low level plans if they misfire. For example, I can be more confident that I will be able to fly to LA than I am that I can fly to LA on any particular flight.

The preceding computation computes the expected value of the plan *in isolation*. But that is not the relevant expected value. The agent may have adopted other plans whose execution will change the context and hence change both the probabilities and values used in computing expected values. Let the agent's *master plan* be the result of merging all of the agent's local plans into a single plan. In deciding whether to adopt a new plan, what is really at issue is the effect that will have on the expected value of the master plan. Changes to the master plan may consist of simultaneously adopting and withdrawing several plans. It is *changes* that must be evaluated decision-theoretically. The value of a change is the difference between the expected value of the master plan after the change and its expected value before the change. This is the *differential expected value of the change.*

In a realistic agent in a complex environment, the master plan may grow very large. It is important to be able to employ simple defeasible computations of expected value. It can be assumed defeasibly that different local plans are *evaluatively independent*, in the sense that the expected value of the combined plan is the sum of the expected values of the individual plans. This makes it easy to compute differential expected values defeasibly. The search for considerations that would make this defeasible assumption incorrect is precisely the same as the search described above for considerations *within a plan* that would change the defeasible computation of its expected value. The only difference is that we look for considerations established by other constituents of the master plan.

Conventional decision theory would tell us to choose a master plan having a maximal expected value. That is at least computationally infeasible in complex domains. There may not even be a maximally good plan. In many domains it may be that we can always improve the master plan marginally by adding more local plans. Instead of maximizing we must *satisfice* — seek plans with positive expected values, and always maintain an interest in finding better plans. A plan is defeasibly adoptable if it has a positive expected value, or if its addition to the master plan increases the value of the latter. The adoption is defeated by finding another plan that can be added to the master plan in its place and will increase the value of the master plan further. So we are always on the lookout for better plans, but we are not searching for a single "best" plan.

6. Conclusions

- An architecture for "anthropomorphic agents" must mimic (but not necessarily duplicate) human rational cognition.
- Practical cognition makes choices based upon information supplied by epistemic cognition.
- Most of the work in rational cognition is carried out by epistemic cognition, *and must be done defeasibly*.
- OSCAR implements a sophisticated system of defeasible reasoning that enables it to deal defeasibly with perception, change and persistence, causation, probabilities, etc.
- Sophisticated agents operating in complex environments cannot plan by using conventional planning algorithms that produce r.e. sets of solutions.
- However, the ideas underlying conventional planning algorithms can be resurrected as *defeasible* principles for reasoning about plans.
- Defeasible principles of deterministic planning can be generalized to produce defeasible principles of decision-theoretic planning.
- In decision-theoretic planning, decisions about whether to adopt new plans (and perhaps to reject previously adopted plans) must be made on the basis of the effect that has on the expected value of the master plan.
- An efficient computation of the expected value of the master plan can be done defeasibly.

References

1. C. Boutelier, T. Dean, and S. Hanks, "Decision-theoretic planning: structural assumptions and computational leverage", *Journal of Artificial Intelligence Research* **10**, forthcoming.
2. Steve Hanks and Drew McDermott, "Default reasoning, nonmonotonic logics, and the frame problem", AAAI-86, 1986.
3. E. J. Horvitz, "Reasoning about beliefs and actions under computational resource constraints", in Laveen Kanal, Tod Levitt, and John Lemmer, editors, *Uncertainty in Artificial Intelligence*, **3**, 301-324, Elsevier/North Holland: Amsterdam, London, New York, 1989.
4. David Israel, "What's wrong with non-monotonic logic?" *Proceedings of the First Annual National Conference on Artificial Intelligence.* 99–101, 1980.
5. N. Kushmerick, S. Hanks, and D. Weld, "An algorithm for probabilistic planning", *Artificial Intelligence* **76**, 239-286, 1995.
6. Edwin P. Pednault, *Toward a Mathematical Theory of Plan Synthesis*, PhD dissertation, Stanford University, 1987.
7. J. Scott Penberthy and Daniel Weld, "UCPOP: a sound, complete, partial order planner for ADL". *Proceedings 3rd International Conference on Principles of Knowledge Representation and Reasoning*, 103-114, 1992.
8. John L. Pollock, *Cognitive Carpentry*, Bradford/MIT Press: Cambridge, 1995.

9. John L. Pollock, "Perceiving and reasoning about a changing world", *Computational Intelligence* **14**, 498-562, 1998.
10. John L. Pollock, "The logical foundations of goal-regression planning in autonomous agents", *Artificial Intelligence* **106**, 267-335, 1998.
11. John L. Pollock and Joseph Cruz, *Contemporary Theories of Knowledge*, second edition, Rowman and Littlefield: Lanham, MD., 1999.
12. Henry Prakken and Gerard Vreeswijk, "Logics for Defeasible Argumentation", to appear in *Handbook of Philosophical Logic*, 2nd Edition, ed. D. Gabbay.
13. Raymond Reiter, A logic for default reasoning. *Artificial Intelligence* **13**, 81–132, 1980.
14. Stuart Russell, "Rationality and intelligence", in A. Rao and M. Wooldridge, *Foundations of Rational Agency,* Kluwer: Dordrecht, 1999.
15. Stuart Russell and Devika Subramanian, "Provably bounded-optimal agents", *Journal of Artificial Intelligence Research* **3**, 1995.
16. Herbert Simon, "Rational choice and the structure of the environment", in *models of Bounded Rationality*, volume 2, MIT Press: Cambridge, 1958.
17. E. E. Smith, C. Langston, and R. Nisbett, "The case for rules of reasoning." *Cognitive Science* **16**, 1–40, 1992.
18 C. B. Suttner and G. Sutcliffe, "The TPTP problem library (TPTP v2.0.0)", Technical Report AR-97-01, Institut für Informatik, Technische Universität München, Germany; Technical Report 97/04, Department of Computer Science, James Cook University, Townsville, Australia, 1997.