This paper appeared in the Proceedings of the 1995 AAAI Fall Symposium on Rational Agency.

Collaboration and Interdependence among Limitedly Rational Agents

Susan L. Epstein
Department of Computer Science
Hunter College and The Graduate School of The City University of New York
New York, NY 10021
epstein@roz.hunter.cuny.edu

Abstract

FORR is a learning and problem-solving architecture that capitalizes upon synergy among a variety of limitedly rational agents. It takes the position that many reasonable but imperfect decision-making agents can, when they agree upon a course of action, quickly make choices that are good enough and will improve with learning. Some FORR agents react quickly and correctly to a small amount of sensed information, others perform highly-restricted search, and still others react heuristically within time constraints. Their collaboration and interdependence on a range of experimental work in two domains are examined here.

Introduction

A *rational agent* does the right thing, that is, behaves in a way that will optimize its performance according to some external standard [Russell and Norvig, 1995]. The agent's actions are based upon its perceptions and its domain knowledge. Because expertise in challenging domains requires extensive, and possibly dynamic, domain-specific knowledge, a rational agent should be able to acquire knowledge as a consequence of its actions, that is, a rational agent should learn. Given perfect knowledge of actions and their outcomes in a domain, it is theoretically possible to simulate perfectly rational behavior, i.e., always to deduce the right thing logically. For real-world problems, however, such computation is usually intractable.

This paper describes a learning and problem-solving architecture called *FORR* (FOr the Right Reasons) that harnesses many different limitedly rational agents to achieve a domain-specific goal. These agents, called *Advisors,* are FORR's "right reasons." Each Advisor epitomizes one simplistic, practical, domain-specific rationale that supports expert decision making in the domain. The Advisors share a common store of *useful knowledge* that is potentially applicable and probably correct.

FORR's thesis is that enough right reasons with enough useful knowledge will eventually do the right thing, i.e., if one can develop and properly coordinate enough Advisors, and give them access to enough reasonably accurate information, a synergy among them will gradually result in better performance without sacrificing efficiency. This paper describes how two FORR-based programs represent knowledge, learn, and coordinate their limitedly rational agents. It also reports and discusses empirical results, related work, and the issues they highlight.

Implementations

FORR's goal is the development of problem-solving expertise. A FORR-based program works on *tasks* (problem-solving experiences where a sequence of actions is intended to reach a desirable world state) within a single *domain* (a set of related problem classes). Two FORR-based programs are used as examples throughout this paper: *Hoyle* takes as its domain two-person, perfect-information, finite-board games [Epstein, 1992a], and *Ariadne* does simulated robot path finding in rectangular, grid-based mazes [Epstein, 1995a]. FORR provides the framework for learning and problem solving, including the representation of knowledge, the construction of experiments, and the collection of data. FORR, Ariadne, and Hoyle are all implemented in Common Lisp.

For Hoyle, a problem class is a game, and a task is a contest at that game. To date, Hoyle has learned to play 18 different games either perfectly or as well as our best external expert programs. This expertise is achieved with the retention of small amounts of new knowledge (no more than .001% of its largest game graphs) and after practice in less than 100 contests. Although it has yet to tackle checkers or chess, many of Hoyle's games have billions of states in their game trees. A challenging decision situation for Hoyle at nine-men's morris is shown in Figure 1.

In Ariadne, each maze (boundaries and obstructions) is a problem class, and a task (pair of locations for the robot and the goal) is a trip through that maze. Ariadne learns to find its way between pairs of locations in a maze represented as a rectangular grid with discrete internal obstructions, like the 20 ∞ 20 maze that is 30% obstructed in Figure 2. A legal move passes through any number of unobstructed locations in a vertical or a horizontal line. The difficulty of a problem is measured as the minimum number of moves required to reach the goal. The robot senses, in any state, its own coordinates, the coordinates of the goal, the dimensions of the maze, and the distance north, south, east, and west to the nearest obstruction or to the goal. The robot does not sense while moving, only before a move. The robot knows the path it has thus far traversed, but it is not given, and does not construct, an explicit, detailed map of the maze. Instead, Ariadne learns descriptive, heuristic information about a particular maze from repeated problem solving in it. This domain is not amenable to traditional AI techniques [Korf, 1990], but after 10 practice trips through a randomly-

This paper appeared in the Proceedings of the 1995 AAAI Fall Symposium on Rational Agency.

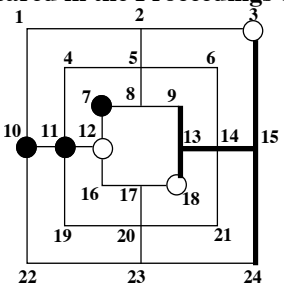

Figure 1: A challenging nine men's morris state black to move.

generated maze, Ariadne quickly solves complex problems requiring as many as 11 right-angle turns *without a map*.

Knowledge and Learning in FORR

Each FORR-based program begins with general knowledge intended only for its set of related problem classes, such as board games or mazes. The user prespecifies a variety of descriptive, domain-dependent information expected to be applicable to every problem class in the domain: problem class descriptions, useful knowledge, and Advisors.

A problem class is defined in FORR as an instantiation of the *problem frame*. FORR's basic problem frame includes domain-independent slots, such as the problem class name. The problem frame is also specialized with domain-dependent slots that identify problem class features known in advance. For example, a game has marker types for each contestant, while a maze has boundary dimensions.

The useful knowledge for each problem class is defined in FORR as an instantiation of the *useful knowledge frame*. FORR's basic useful knowledge frame includes domain-independent slots such as average task length. The useful knowledge frame is also specialized with domain-dependent slots that identify problem class features whose values are not known in advance (unlike those in the problem frame)

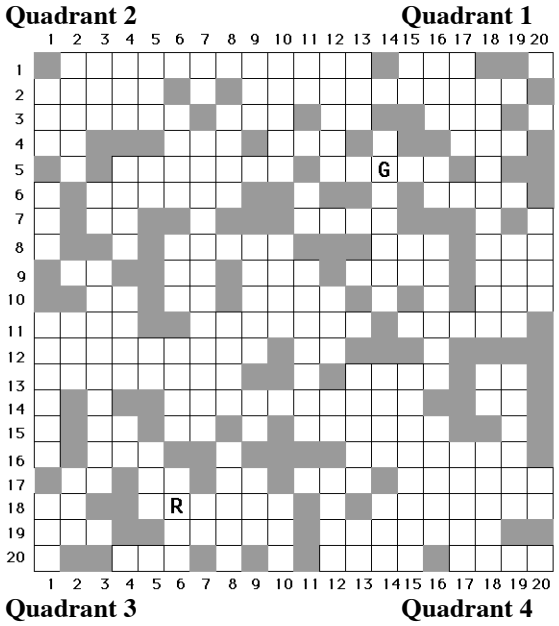

Figure 2: Ariadne's robot R must move through the grid to the goal G in unidirectional steps through unobstructed locations *without* the map shown here.

but worth learning. For example, Hoyle learns good openings and Ariadne learns dead-ends. Useful knowledge may be thought of as the right questions to ask about a problem class. Each item of useful knowledge has a relative learning schedule (e.g., after a decision or some set of decisions) and an appropriate learning algorithm. The slot specifies what to learn, while its associated learning algorithm specifies how to learn it. Any learning method may be employed in these algorithms. Hoyle, for example, learns some items of useful knowledge inductively, others deductively, and one with a graph-oriented variation of EBL [Epstein, 1990].

An Advisor is an agent that epitomizes a domain-specific but problem-class-independent, decision-making rationale, such as "minimize the other contestant's material" or "get closer to your destination." An Advisor is implemented as a time-limited procedure whose input is the current state of the world, the current permissible actions from that state, and any learned useful knowledge about the current problem class. Each Advisor outputs any number of *comments* that support or discourage permissible actions. A comment lists the Advisor's name, the action commented upon, and a *strength*, an integer from 0 to 10 that measures the intensity and direction of the Advisor's opinion. For example, one comment for the state in Figure 2 from an Advisor that advocates small steps would be <Plod, (17, 6), 8>. Although there are no constraints on the nature of the comment-generating procedures themselves, a FORR-based system is intended to sense the current state of the world and react with a rapid computation, i.e., to eschew extensive search.

FORR models the transition from general expertise to specific expertise as the acquisition of useful knowledge [Epstein, 1994a]. From its experience, a FORR-based program specializes its domain-wide knowledge gradually for each problem class. The same Advisor, when confronted with the same state of the world, may comment differently as useful knowledge evolves. For example, Hoyle's Open relies upon useful knowledge of good game openings, and Ariadne's Quadro relies upon its knowledge of gates, transitions between quadrants in the layout. Increasingly accurate and complete useful knowledge enhances the value of each Advisors' comments.

How Rational Agents Work Together

Not all rational agents are expected to be equally important. FORR partitions its Advisors into three tiers. Tiers 1 and 2 are fundamentally reactive, while those in the intermediate tier 1.5 interact overtly with their environment. Advisors in Tier 1 reference only correct useful knowledge and their comments are trustworthy. They sense the current state of the world and whatever correct useful knowledge they have about the problem class; if they make a decision, it must be fast and correct. Hoyle has a tier-1 Advisor called Panic based upon the rationale "if the other contestant has an immediate a winning move, block it." Ariadne has one called No Way based upon the rationale "do not enter a dead-end unless it could contain the goal."

In contrast, Advisors in Tier 2 are not necessarily correct in the full context of the state space. Each of them epitomizes a heuristic, private system of belief that can

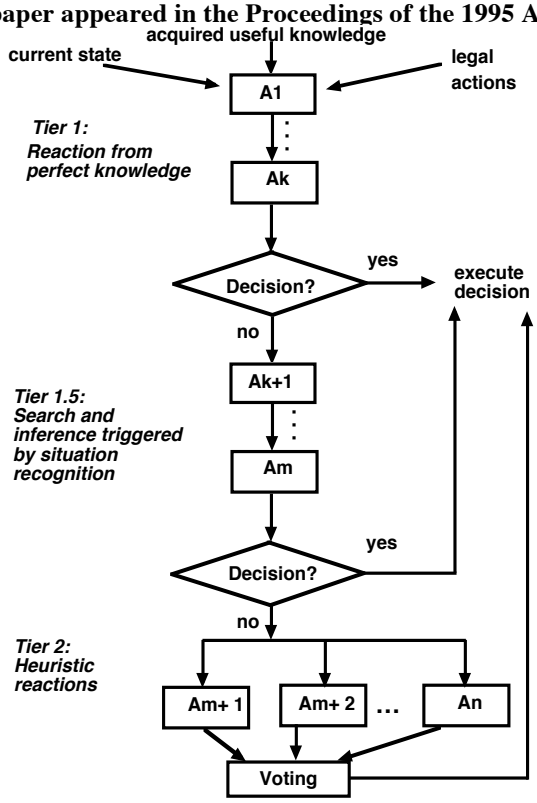

Figure 3: How FORR agents collaborate.

make a valid argument for or against one or more actions. Neither their reasoning processes nor the useful knowledge on which they rely is guaranteed correct, and each has only a limited time for computation. Hoyle has a tier-2 Advisor called Material that advocates "maximize the number of your pieces on the board and minimize those of the other contestant." Ariadne has one called Plod that represents "take a one-unit step, preferably toward the goal."

Advisors in tier 1.5 do highly-constrained, knowledge-based, heuristic search. Each has a reactive trigger and a search procedure that generates and tests a highly-constrained set of possible solution fragments. The *trigger* signals recognition that the Advisor may be pertinent to the current situation. The search procedure attempts to construct a *solution fragment*, a sequence of recommended decisions rather than a single reactive one, a digression from the "sense-compute-execute" loop. Ariadne, for example, has a tier-1.5 Advisor called Wander. Wander triggers when the robot's recent locations represent a relatively small fraction of those in the maze. Wander's search procedure can investigate as many as eight L-shaped paths (by moving one longest step in each direction and then testing for possible second steps) before it returns one as a solution fragment.

Figure 3 shows how FORR effects collaboration among its three tiers to make decisions. Tier-1 Advisors are consulted in a predetermined, fixed order. Each Advisor may have *absolute authority* to make a decision alone, or *veto power* to eliminate a legal action from any further consideration. Only when the first tier of a FORR-based system fails to make a decision does control default to tier 1.5. Tier 1.5 is prioritized too; each Advisor in turn is given the opportunity to trigger. Once a tier-1.5 Advisor triggers, control is ceded to its search procedure with limited time to formulate and test possible solution fragments. The first solution fragment returned by a tier-1.5 Advisor is executed and then, regardless of the outcome, control is returned to tier 1. If no tier-1.5 Advisor triggers, or none produces a solution fragment, the second tier will make the decision. All tier-2 Advisors have an opportunity to comment before any decision is made. The decision they arrive at is the action with highest total strength; this represents a consensus of their opinions. (A tie is broken by random selection.) For example, in Figure 1 it is Hoyle's turn to place a marker on the board, playing black, and white is in the midst of a fork (more than one threat, indicated by the bold lines, to make three in a row and thereby capture a black marker) that would give it a substantial advantage. Although a conventional game-playing program would require 5-ply search to detect the fork, Hoyle's Advisors Greedy and Mobility selected a very strong move here (to 24) that not only defeated the fork but also began another fork for black with only 2-ply lookahead.

At no time do FORR's Advisors actually dialogue with each other or with the *external expert* program available in some domains. (Hoyle, for example, learns against a different, hand-crafted, external expert program for each game.) If an external expert exists, FORR's Advisors can only observe its behavior, not query it. Implicit interaction among Advisors is the sharing of a common useful knowledge base. Explicit interaction among Advisors comes when one with absolute authority prevents the others from commenting at all, when one with veto power prevents the support of an action by the others, when a tier-1.5 Advisor constructs a solution fragment before any tier-2 Advisors ever comment, or when a group of tier-2 Advisors' comments combine to override the others' expressed opinions.

Empirical Lessons on Limited Rationality

In a series of ablation experiments with Ariadne, we have demonstrated an important synergy among correct reactivity in tier 1, heuristic search in tier 1.5, and heuristic reasoning in tier 2 [Epstein, 1995a]. Correct reactivity offers the commonsense inherent in any problem solving task, while heuristic reasoning offers quick expert rules of thumb that try to avoid search. Ariadne shows, however, that on occasion to react well one needs to know more about the problem space, i.e., to search it. Four of Ariadne's five tier 1.5 Advisors are triggered by lack of problem-solving progress. Effectively, when the available useful knowledge is inadequate to support the heuristic reasoners, Ariadne's *tier 1.5 sends out search agents that, as a side effect, learn*. Not only do these agents attempt to extricate the robot, but they also cache any useful knowledge they acquire during search. After a tier-1.5 Advisor constructs a solution fragment, all the Advisors have access to more knowledge and may be in a better position to use it. For example, Ariadne's tier-1.5 Advisor Wander puts the robot where all the tier-2 Advisors are more likely to make new, constructive comments. In turn, heuristic reasoning agents often create situations in which heuristic search agents can produce important solution fragments. For example, Ariadne's tier-2 Advisors Goal Row and Goal Column push the robot into situations

where Roundabout, a tier-1.5 Advisor for circumnavigating walls, can trigger.

A FORR-based program begins learning in a problem class with a host of Advisors presumed to be limitedly rational in the domain, but with no indication of their relative worth in the specific problem class. They need to be *validated*, i.e., to have their usefulness and trustworthiness confirmed in the current context. Despite the intended generality of FORR's rational agents, their degree of applicability varies from one problem class to another. *Relevance validation* attempts to eliminate Advisors that make no contribution in a particular problem class. FORR monitors the participation of each Advisor. Those that never comment (like Mobility in tic-tac-toe) are periodically proposed to a human supervisor as possibly irrelevant. With confirmation, FORR reduces computational overhead by no longer consulting them for that problem class [Epstein, 1994b]. Human monitoring is needed thus far because some Advisors are "slow starters," i.e., need enough useful knowledge to comment.

FORR addresses domains where a sequence of good decisions is required to achieve the goal, but success or failure is not readily attributed to any particular decision. Credit and blame assignment are therefore directed to the Advisors responsible for good and bad decisions, rather than to the decisions themselves. In addition, some programs, such as Hoyle, learn pattern-based, problem-class-specific Advisors that summarize experience and should be gradually integrated into Figure 3 as they prove their worth [Epstein and Gelfand, 1995]. Although these learned Advisors are carefully constructed and filtered, they, too, need validation. *Significance validation* attempts to fit the behavior of a FORR-based program to an observable external expert agent and to avoid the behavior of a random one. If there is an external model of expertise, such as the hand-crafted programs Hoyle learns against, then the comments of the tier-2 Advisors and the pattern-based Advisors are compared with the decisions of a random agent and of the external expert program. Advisors that underperform the random agent or contradict the external expert agent are blamed, and those that consistently agree with the external expert agent are rewarded [Epstein, 1994b]. In this way FORR learns problem-class-specific weights to emphasize the strengths of expert-like Advisors and to filter out weaker ones .

We have experimented with a variety of agents in both Hoyle and Ariadne. Some critical mass of agents appears essential to success in these domains, particularly in the second tier. Prior to learning problem-class-specific Advisors, Hoyle has 23 agents, 7 in tier 1 and 16 in tier 2. Ariadne has 21 agents, 2 in tier 1, 5 in tier 1.5, and 14 in tier 2. We have also tested agents that make random decisions. In a challenging game, such an agent playing alone will lose every contest; in a maze such an agent traveling alone fails to solve any of the more difficult problems.

A perfectly rational external agent is an inadequate model from which to develop a robust, reasonable one. If there is an external model of expertise in the environment, such as another contestant at a two-person game, then the nature of that model has a substantial influence on the speed of FORR's learning and the quality of the expertise it eventually develops [Epstein, 1994c]. This is not only because some FORR agents rely upon that external model as a paradigm (as when Hoyle's Open replicates an opening it has seen an expert play), but also because the external model effectively guides the learner to the most important portions of the search space. Care must be taken, however, to permit the learner to explore on its own as well.

Agents reason better from explicit concepts. Without explicit conceptual knowledge, a FORR-based program's ability to develop expertise is substantially reduced [Epstein, 1992b]. Learning, at least for Hoyle, is reduced to rote memorization, destined to be intractable in a large search space. Learning is also essential to Ariadne's ability to solve its more difficult problems; on the easier ones the program fares well enough without it [Epstein, 1995b].

Discussion

FORR is a *satisficing* architecture, one that constructs "good enough" decisions; it is prepared to sacrifice theoretical optimality for evolving expertise. FORR's Advisors are suboptimal in several ways. Time and procedural constraints on search make them *limitedly rational*. For example, a "rational" game player would simply search the entire game tree, and a "rational" robot would exhaustively search its maze. Advisors' *strategic rationality* (support and discouragement of particular actions) is restricted by the narrowness of their individual perceptions. For example, Ariadne's Plod only addresses the direction of a step, with no concern for the robot's distance from the goal. Advisors' *epistemic rationality* is questionable because they rely on useful knowledge without any guarantee of consistency or deductive closure. One might well argue that these are *reasonable,* rather than rational, agents.

We have yet to encounter, however, any serious performance difficulties as a result. FORR's robustness, we believe, is attributable to the multiplicity of reasonable agents and reasonable knowledge acquisition methods it has at its disposal. In Ariadne, for example, an extent is useful knowledge about a bounding rectangle for some area in a maze. Among the ways an extent may be learned are as a bottle and as a chamber. Each is a heuristic description for a region in which the robot has once been confined. If the robot is trapped there again, Advisors that reference either or both kinds of extents can help to extricate it.

Reasonable agents also provide explanations people understand. For example, Ariadne's tier-2 voting in a particular situation might be interpreted as "this is a good choice because it gets me closer to the goal, is a large step, and takes me to a location I have never been in on this trip." Similarly, Hoyle's refusal to make a particular move might be argued as "this is a poor choice because it reduces my potential mobility on my next move and is symmetric to a move I once explored with questionable results."

If expertise in a given domain can be represented as set of condition-action rules with a control structure based upon absolute authority and veto power, FORR can implement it as a collection of tier-1 Advisors. For most challenging domains, however, such perfect knowledge is unavailable.

One traditional AI alternative is to incorporate additional, heuristic rules into a system and then attempt to sequence and tune them. FORR's tier-2 Advisors are actually sets of heuristic rules. The system is spared the burden of ordering or balancing those rules, simply because they all vote together. Thus tier 2 forms an action-value system, where actions are evaluated not on some absolute scale, but with respect to each other. The preference function is the combined vote from all the tier-2 Advisors.

Hoyle and Ariadne are both faced with uncertainty, albeit for different reasons. Hoyle could theoretically explore the entire search space, but for most games that would take too long, particularly without game-specific hardware like that used by state-of-the-art chess machines [Anantharaman, et al., 1990; Berliner and Ebeling, 1989]. Ariadne, however, is hampered by its limited sensing ability. It cannot, for example, see around corners or detect the length of a wall unless it acts. So Hoyle, in some sense, chooses not to know, while Ariadne is helplessly ignorant. Both systems, however, successfully deal with uncertainty by satisficing.

FORR's global approach to a domain with a set of independent knowledge sources is reminiscent of a distributed system. Each knowledge source is implemented as an agent (an Advisor) with a common goal but is heuristic, and its comments may be inaccurate. FORR's agents do not negotiate; they act together because the control structure of Figure 3 forces their collaboration [Levesque, et al., 1990]. This coordination relies upon a high-level strategic plan for advice sharing, similar in spirit to that of Corkill and Lesser [Corkill and Lesser, 1983]. The plan, however, is partially predetermined and partially learned, and agents have carefully delineated interactions. Each FORR agent spends most of its time in computation rather than communication, as do those in DARES [Conry, et al., 1990]. Unlike DARES, however, lack of direct communication frees FORR's individual agents to use powerful, even idiosyncratic, knowledge representations that support efficient reasoning from a particular viewpoint.

Issues and Implications

We foresee several important issues in our ongoing work. First, particularly with the automated acquisition of new Advisors, the assignment of Advisors to tiers and their prioritization within those tiers becomes an issue. This is more difficult in domains such as Ariadne's, which have no external model as a standard of good performance. Second, the contribution of an Advisor in FORR should be a function of its computational cost, its reliability, and, perhaps, its transparency. All of these may be problem-class-dependent and should be learned. Finally, automaticity is the gradual transition from high-level reasoning to rapid, compiled computation. If a new, learned, reactive Advisor compiles out knowledge that other, slower Advisors have, should it replace them? precede them? compete with them?

Hoyle's pattern-based, problem-class-specific Advisors are part of our research on the automated generation and application of limitedly rational agents. Current research also includes the identification of domain-independent Advisors that FORR generates and provides, just as it provides the basic slots in the problem frame. Victory is such an Advisor.

FORR is not ideal in every domain. Experience must be readily available and inexpensive, and the domain must be able to tolerate failure. An external model, while not essential, is helpful. The user must also be able to express reasonably correct and complete information about the domain through a problem class definition, Advisors, useful knowledge, and learning algorithms. Finally, satisficing must meet the user's needs.

For many intractable real-world problems, however, a suboptimal solution is acceptable, and there is evidence that people reason this way [Biswas, et al., 1995; Crowley and Siegler, 1993; Ratterman and Epstein, 1995]. Once the domain is well represented, FORR has proved to be a robust and adaptive approach to limitedly rational agency.

References

Anantharaman, T., Campbell, M. S. and Hsu, F.-h. 1990. Singular Extensions: Adding Selectivity to Brute-Force Searching. *Artificial Intelligence* 43 (1): 99-110.

Berliner, H. and Ebeling, C. 1989. Pattern Knowledge and Search: The SUPREM Architecture. *Artificial Intelligence* 38 (2): 161-198.

Biswas, G., Goldman, S., Fisher, D., Bhuva, B. and Glewwe, G. 1995. Assessing Design Activity in Complex CMOS Circuit Design. In *Cognitively Diagnostic Assessment,* ed. P. Nichols, S. Chipman and R. Brennan. Hillsdale, NJ: Lawrence Erlbaum.

Conry, S. E., MacIntosh, D. J. and Meyer, R. A. 1990. DARES: A Distributed Automated REasoning System. In *Proceedings of the Eighth National Conference on Artificial Intelligence,* 78-85. AAAI Press.

Corkill, D. D. and Lesser, V. R. 1983. The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence,* 748-756.

Epstein, S. L. 1990. Learning Plans for Competitive Domains. In *Proceedings of the Seventh International Conference on Machine Learning,* 190-197. Ed. B. W. Porter and R. J. Mooney. Morgan Kaufmann.

Epstein, S. L. 1992a. Prior Knowledge Strengthens Learning to Control Search in Weak Theory Domains. *International Journal of Intelligent Systems* 7: 547-586.

Epstein, S. L. 1992b. The Role of Memory and Concepts in Learning. *Minds and Machines* 2: 239-265.

Epstein, S. L. 1994a. For the Right Reasons: The FORR Architecture for Learning in a Skill Domain. *Cognitive Science* 18 (3): 479-511.

Epstein, S. L. 1994b. Identifying the Right Reasons: Learning to Filter Decision Makers. In *Proceedings of the AAAI 1994 Fall Symposium on Relevance.,* 68-71. AAAI.

Epstein, S. L. 1994c. Toward an Ideal Trainer. *Machine Learning* 15 (3): 251-277.

Epstein, S. L. 1995a. On Heuristic Reasoning, Reactivity, and Search. In *Proceedings of the IJCAI-95,* in press.

Epstein, S. L. 1995b. On the Roles of Search and Learning in Time-Limited Decision Making. In *Proceedings of the 17th Annual Cognitive Science Conference,* 568-573. Ed. J. D. Moore and J. F. Lehman. Lawrence Earlbaum.

This paper appeared in the Proceedings of the 1995 AAAI Fall Symposium on Rational Agency.

Epstein, S. L. and Gelfand, J. 1995. Learning New Spatially-Oriented Game-Playing Agents through Experience. In *Proceedings of the Seventeenth Annual Cognitive Science Conference,* 562-567. Ed. J. D. Moore and J. F. Lehman. Lawrence Earlbaum.

Korf, R. E. 1990. Real-Time Heuristic Search. *Artificial Intelligence* 42 (2-3): 189-211.

Levesque, H. J., Cohen, P. R. and Nunes, J. H. T. 1990. On Acting Together. In *Proceedings of the Eighth National Conference on Artificial Intelligence,* 94-99. AAAI Press.

Ratterman, M. J. and Epstein, S. L. 1995. Skilled like a Person: A Comparison of Human and Computer Game Playing. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society,* 709-714. Ed. J. D. Moore and J. F. Lehman. Lawrence Erlbaum.

Russell, S. and Norvig, P. 1995. *Artificial Intelligence - A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall.