VICTORIA UNIVERSITY OF WELLINGTON

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

Foundations of *Foundations of Artificial Intelligence*

Ashwin Ram
Eric K. Jones

Technical Report CS-TR-94/3
March, 1994

Abstract

We review the book *Foundations of Artificial Intelligence* (ed. David Kirsch, MIT Press, Cambridge, MA, 1993) and present a framework for the analysis of theories of artificial intelligence.

Publishing Information

To appear in *Philosophical Psychology*.

Author Information

Ashwin Ram is an Assistant Professor in the College of Computing of the Georgia Institute of Technology, and an Adjunct Professor in the School of Psychology. He received his B.Tech. in Electrical Engineering from the Indian Institute of Technology, New Delhi, in 1982, and his M.S. in Computer Science from the University of Illinois at Urbana-Champaign in 1984. He received his Ph.D. degree from Yale University for his dissertation on "Question-Driven Understanding: An Integrated Theory of Story Understanding, Memory, and Learning" in 1989. His research interests lie in the areas of machine learning, natural language understanding, explanation, and cognitive science, and he has several research publications in these areas. Dr. Ram is a co-editor of a book on Goal-Driven Learning, forthcoming from MIT Press/Bradford Books. He is a member of the editorial boards of the Journal of the Learning Sciences and the Journal of Applied Intelligence, and an associate of Brain and Behavioral Sciences.

Eric Jones is on the academic staff of the Department of Computer Science at Victoria University of Wellington, New Zealand. He received his Ph.D. in artificial intelligence from Yale University in 1992 for his dissertation entitled "The Flexible Use of Abstract Knowledge in Planning." He also has degrees in mathematics and geography from the University of Otago, New Zealand. His research interests include intelligent database retrieval, case-based reasoning, natural language processing, and machine learning.

Foundations of Artificial Intelligence presents a number of chapters from major players in artificial intelligence (AI) that discuss fundamental assumptions underlying the dominant approaches to AI today. Perhaps the best parts of the book are the critiques: each chapter is followed by an in-depth critique that evaluates the utility of those assumptions in pursuing the goal of AI.

But what is the goal of AI? Although several chapters propose definitions of the AI enterprise, there seems to be little agreement even at this fundamental level. Kirsh discusses the following definition in his introduction:

> A theory in AI is a specification of the knowledge underpinning a cognitive skill. (p. 5)

While there appears to be a broad consensus (with some dissension from Brooks) that knowledge specification is an important part of the practice of AI, there seems to be little agreement that knowledge specification by itself constitutes a theory in AI. Indeed, while Lenat and Feigenbaum take this position seriously, Nilsson focusses on the language for the specification of such knowledge (rather than the knowledge itself); Hewitt on communication between agents; Rosenbloom, Laird, Newell, and McCarl on architectural issues in lieu of knowledge; and Brooks eschews explicit representations of knowledge altogether.

This lack of consensus is both the principal strength and weakness of the book. AI as a field is still in its formative stages, and the diversity of approaches and methods is important to its development. This diversity is well represented by the book. The individual chapters are for the most part interesting and well written, and the debate between conflicting viewpoints is lively and informative. On the other hand, the lack of an integrating view makes the book as a whole hard to follow. Different chapters often directly contradict one another, and when the dust settles, the reader is left unclear as to what the AI enterprise really is.

If the field of AI can be meaningfully said to have foundations—and we believe that it can—it should be possible to identify some fundamental level of agreement as to what those foundations are. Kirsh's introduction identifies a number of key issues for debate, but does not attempt to resolve them. While we do not expect agreement at the level of particular methods or approaches, it is appropriate for a book about foundations to identify points of agreement about the nature of AI theories and about the kinds of problems of science and engineering that AI theories should address. Unfortunately, the book does not make explicit what, if any, points of agreement exist between the authors, nor does it present an integrative viewpoint that helps put the individual chapters into a common perspective.

Rather than reviewing individual chapters, therefore, let us attempt to discuss the points of agreement and differences of focus among the various authors. We will then suggest an integrating framework in which to fit the authors' contributions. It will become clear that many apparent disagreements between the authors are differences only in focus or research interests, not disagreements on foundational issues.

Points of Agreement—Almost

A central issue in the debate about foundations concerns the role of knowledge representation in AI, and the form that such representations should take. At first it might seem that there is a fundamental split in the AI community between researchers like Nilsson, Lenat, and Feigenbaum who assert the centrality of explicit representations, and researchers such as Brooks who seem to deny a role for representation altogether. However, we believe this split to be more apparent than real, the title of Brooks' chapter notwithstanding—*Intelligence without representation.*

To understand why we believe this to be the case, it is useful to distinguish two very different ways that representations are used in AI. First, representations can be used at the level of computational models to specify the knowledge required to solve some class of tasks. Second, many implemented systems manipulate explicit representations. It is important to emphasize that explicit representations can play an important theoretical role in knowledge specification without necessarily appearing in an implementation.

To use the terminology of Dennett (1978), we must distinguish between taking a design stance towards an intelligent system and taking an intentional stance. Does the system actually have representations (design stance) or does it merely act as if it might (intentional stance)? In the latter case, the system can still usefully be modeled using a representational formalism.

When these differing roles for representation are kept in mind, much apparent disagreement about representation dissolves. For example, Brooks objects to explicit representations because he believes that they encourage the researcher to "cheat" by supplying programs with all their abstractions. He claims that AI researchers partition problems into an easy "AI" component that their programs can handle, and a hard "non-AI" component of specifying suitable abstractions that only the programmer can. Explicit representations are bad because their primary function is to support hand-crafting of these abstractions: "But [the process of] abstraction is the essence of intelligence and the hard part of the problems being solved" (p. 143). Abstraction as currently used in AI, Brooks concludes, is "a mechanism for self-delusion" (p. 142).

But Brooks also supplies his creatures with all of their abstractions:

> individual layers extract only those *aspects* ... of the world which they find relevant. (p. 148)

"Aspects" are simply abstractions by another name, and are specified by a human designer just like the declarative representations of traditional AI systems. Indeed, upon reflection, it becomes clear that the issue of declarative representations is largely irrelevant to Brooks' concerns. The important theoretical issue is not how aspects or abstractions should be encoded inside the machine—using explicit representations or not—but rather how abstractions should be chosen by the machine's designers. Brooks' fundamental contribution is to show that the activity of building real robots and testing them in unrestricted environments is a good way to go about discovering useful abstractions for intelligent systems. When Brooks takes an intentional stance towards his programs, he is as committed to representation as the rest of us.

In summary, all of the authors agree that knowledge is necessary—lots of it. Even Brooks can be seen as adhering to this view: although Brooks is opposed to centralized representations and detailed world models, activity-specific representations of selected aspects of the real world play a central role in his systems.

Indeed, an emphasis on real-world constraints appears to be another general point of agreement. Lenat and Feigenbaum, for example, are very concerned with the brittleness of AI systems when faced with real-world problem-solving situations, although their particular approach to this problem diverges from that of Brooks in important ways. More generally, all contributors agree that AI should get busy building real systems that attempt difficult and realistic tasks. No more toy stuff!

All of the authors except Brooks accept the importance of mechanisms for inference and learning. And there is no reason to suppose that Brooks will continue to reject such mechanisms once he attempts to scale up his systems to cope with a wide range of cognitive tasks. Where mechanisms are discussed, however, there are some important differences in approach. Rosenbloom, Laird, Newell, and McCarl, for example, propose a very small number of general-purpose mechanisms for a large range of cognitive tasks, such as universal subgoaling for problem-solving and chunking for learning; other authors lean more towards a larger and more eclectic collection of cooperating, special-purpose mechanisms.

While it is evident that reasoning systems must reason, it is less evident that reasoning systems must also act. Brooks and others have led a recent broadening of emphasis in AI research towards the role of perception and action in intelligence, advocating systems that are more than "armchair reasoners." The authors in this book—aside from possibly Lenat and Feigenbaum—accept the need for general mechanisms for acting in the real world.

Somewhat to our surprise, although the contributions acknowledge the need for learning, most of them do not actually discuss mechanisms that might perform learning. To quote Schank (1987):

> Learning is, after all, the quintessential AI issue. [...] I will now give a definition of AI that most of our programs will fail: AI is the science of endowing programs with the ability to change themselves for the better as a result of their experiences (pp. 63–64).

In his introduction, Kirsh arrives at a similar conclusion: "Much of cognition cannot be studied independently of learning" (p. 27). However, apart from the Soar team, none of the authors in this book appears to actually be working on learning; we were also disappointed that there is no machine learning chapter in this book. Surely any general theory of intelligence (as opposed to a somewhat smart program) must have a place for learning?

Towards a Framework for Consensus

To come to some sort of consensus regarding the foundations of AI, we first need a working definition of AI. AI, in our view, is concerned with developing theories of intelligence (and, often, theories of human intelligence, although as Norman points out the integration of artificial intelligence with cognitive science is not as straightforward as it might appear). Research in AI proceeds by constructing detailed computational models of intelligent behavior; it is this emphasis on detailed computational models that distinguishes AI from related fields such as cognitive psychology and philosophy of mind. By "intelligent behavior" we mean tasks that are commonly regarded by people as requiring intelligence, such as planning, natural language understanding, diagnosis, and commonsense reasoning. Computational models are then implemented in computer programs and tested through systematic evaluation of these programs.

A crucial issue that any consensus on the foundations of AI must address is the relation between AI theories and programs. To help develop and test their computational models, AI researchers also build computer programs that instantiate the model. These programs must do something: they must perform a task or tasks in a given range of problem domains.

Some of the seeming disagreement between the authors stems from differing views about the goal of the AI enterprise. Is AI primarily about building intelligent systems or about developing theories of intelligence? Brooks, for example, focuses on building embodied programs that perform some real-world task. Rosenbloom *et al.*, in contrast, aim to "understand the functionality required to support general intelligence" (p. 290); to them, computer programs serve primarily as tests of a theory's adequacy. Lenat and Feigenbaum are concerned with problems arising from the inadequacies of implemented AI programs. Nilsson, on the other hand, deemphasizes considerations of tasks and computational feasibility, and focuses primarily on the role of logic in formalizing knowledge.

So what *is* the role of the program in AI? The product? A tool to test and refine theories? What constitutes a satisfactory demonstration of a theory? How should theories be evaluated? While *Foundations of Artificial Intelligence* provides no single answer to these questions, it suggests that building intelligent programs is an important part of the AI enterprise. Historically, however, many AI programs have not been intended as examples of complete intelligent systems, but only constitute a single component of a hypothetical larger system that never seems to get built. Brooks, Lenat and Feigenbaum, and Rosenbloom *et al.* convincingly argue that we should scale up our programs to the point where this is no longer true. We also agree with Hewitt and Gasser that the task environment of these agents must include other agents.

It is important to emphasize, however, that programs are not by themselves theories of AI. To quote from Boden's (1977) preface:

> AI is not the study of computers, but of intelligence in thought and action. Computers are its tools, because its theories are expressed as computer programs that enable machines to do things that would require intelligence if done by people.

It is the theory that is the final product of an AI research endeavor. In some cases, programs are useful technological artifacts in their own right, but then the theory should provide a good account of the scope and limits of the engineering methodology that the program exemplifies. More generally, programs should assist the process of theory development by supporting a cycle of hypothesis formulation, testing, and revision. This leads to the obvious question: what should a theory of AI consist of, such that a program can be used to test it? Somewhat surprisingly, none of the authors of *Foundations of Artificial Intelligence* provides a satisfying answer to this question.

An AI theory cannot be primarily a specification of knowledge, as Nilsson and other logicists at times seem to suggest. If programs must carry out some real task—and the authors in the book seem to agree they should—then knowledge needs to be organized so that it is available in appropriate circumstances. Several decades of AI research have demonstrated that the problem of knowledge organization is far from trivial.

Moreover, specifications of knowledge cannot be divorced from the way that the knowledge is used: knowledge needs to be situated, as both Birnbaum and Smith convincingly argue. It would be more than a little surprising if a system such as CYC should turn out to be able to solve some non-trivial real-world task without a major redesign. CYC is not physically embodied and has no task domain; without the constraints of a real-world environment or task, the design process seems too unconstrained to be likely to produce something very useful.

An AI theory needs a representation language—a formalism in which to describe the knowledge required for a task. However, an AI theory cannot be only a language (logic or otherwise); that is like saying that calculus is a theory of physics. Hayes (1985) argues, for example, that large-scale formalizations of knowledge of the everyday world are of basic importance to artificial intelligence and cognitive science: "The scientific questions of interest are to do with [the interrelationships between concepts in a theory], not the idiosyncrasies of any particular notation for recording them" (p. 34). Likewise, Davis (1990) distinguishes between the "domain model" and the "axiomatic system" used to represent it (p. 7). Nilsson discusses logic, the representational formalism of choice for many AI researchers, and the importance of sound inference, but what are those inferences about? How should the inferences be carried out? What real-world distinctions should representations encode? And can existing logical formalisms adequately capture the impact of the context in which agents are situated?

Hewitt addresses some of these concerns in the context of his framework for analysis of distributed and multi-agent systems. He discusses methods for inference and communication that presumably operate over the knowledge encoded in the chosen representational language.

But an AI theory is more than such a framework—Hewitt ignores issues of content. What do the agents know? What do they talk about? What do they do? How do they interact with one another? Indeed, many of the authors in the book focus on architectures or formalisms rather than on content. Rosenbloom and his colleagues, for example, discuss a general-purpose architecture in which knowledge and inferential mechanisms can be encoded. But while an architecture is an important part of an AI theory, an AI theory cannot be just an architecture (Soar or otherwise). Again, what does an agent know? What is the range of tasks and contexts in which that knowledge is applicable? What are the algorithms for reasoning, learning, perception, and acting that are to run on that architecture?

An AI theory cannot be merely a program or implemented system, however cleverly engineered. For example, Brooks' creatures are interesting not just because they work (which is the goal that drives his research), but because Brooks also provides an abstract specification of the kinds of knowledge that they need, and provides good arguments for layered designs that decompose tasks by activity rather than by function.

Although the chapters in the book span the range of AI problems (except possibly learning) and approaches (except connectionism), many of the authors seem to be talking past one another. The concerns addressed in the individual chapters are certainly important; however, each individual chapter, while informative with respect to its chosen topic, addresses only one or two aspects of the AI enterprise, and few hints are given as to how one might combine different perspectives to develop a larger testable theory of intelligence (as opposed to just a piece of one). To this end, we now attempt to draw together the disparate threads that the book presents, and synthesize a general conception of AI theories to which each of the authors contributes a part.

The Elements of an AI Theory

A general theory of intelligence will span many tasks and domains, but such a theory appears to be a long way off. Most AI research necessarily focuses on a limited range of intelligent behaviors, and proceeds by developing detailed computational models of these behaviors. Computational models are (or should be!) accompanied by implemented systems that exemplify the model and test its adequacy. Systematic experiments on these systems are then used to evaluate the theoretical claims.

An implementation needs a task, a domain, and an environment. By "task" we mean a general class of activity such as planning, diagnosis, or language comprehension. The domain is the subject matter of a task. A program whose task is language comprehension, for example, might have as its domain news stories about terrorism in Latin America. The environment of an embodied agent is some part of the real world. For example, Brooks has constructed a robot called Herbert that wanders around collecting empty Coca-cola cans. Here the task is robot planning, the domain is collecting Coca-cola cans, and the environment is the MIT AI lab. Disembodied programs also have environments. The environment of a language comprehension program, for example, might be a large corpus of stories and a standard set of questions to ask the program that test whether it adequately comprehends a story.

There is vast gap between a program that implements a specific task in a given domain and a general theory of intelligence. It is therefore crucial that a program be accompanied by an analysis that abstracts away from the details of the program and explains how the program exemplifies a theory of some aspect of intelligent behavior. The theory should specify just what class of tasks and range of problems are within its scope, and specify how to evaluate whether or not the theory succeeds. Many researchers fail completely to provide such analyses; indeed, in a survey of papers accepted to the AAAI-90 conference, Cohen (1991) discovered that only 43% of the papers that described implemented systems report any kind of analysis of their contributions. Even of the papers that do describe evaluatory experiments, very few go beyond evaluating the programs to analyzing the scientific claims that the programs were written to demonstrate. In what follows, we present a framework for formulating suitable analyses.

In our view, a theory of intelligent behavior should have a descriptive part and an explanatory part. The descriptive part specifies the computational mechanisms of the theory, and makes clear how the program instantiates those mechanisms. Computational mechanisms can be described under the following headings:

- *Knowledge*: both the content of the relevant knowledge and the representation language used to express that knowledge.
- *Processes*: the algorithms or mechanisms that produce the intelligent behavior.
- *Architecture*: the "cognitive architecture" on which the algorithms execute.
- *Machine architecture*: the physical hardware, if this happens to be theoretically relevant.

A theory of intelligent behavior also has an explanatory part, which justifies the computational mechanisms of the theory by explaining the way in which they are a good account of the behavior. The explanation provides a functional or teleological basis for the design decisions underlying the computational model, such as the choice of representational primitives and formalisms, and architectural and algorithmic commitments. The explanation should also make clear how the computer implementation exemplifies this account.

A wide range of illuminating explanations is possible, and the kind of explanation that a theory aims to provide depends heavily on the school of AI thought to which a researcher subscribes. Here are some kinds of explanations:

- *Psychological*: explanations that appeal to data from human psychology. For example, Soar exhibits the power law of learning, and in so doing provides an explanation of that property of human learning.
- *Computational*: arguments from computational complexity or other properties of algorithms. For example, Allen's (1983) path consistency algorithm for temporal constraint propagation is justified on the basis of its polynomial time complexity.
- *Ecological*: arguments that a certain kind of knowledge or architecture is best suited for a class of tasks given a particular kind of environment. Chapman and Agre's (1987) arguments for indexical-functional representations and Brooks' arguments in favor of his subsumption architecture are good examples of this approach, as are teleological arguments for representational choices for a given class of tasks and domains (e.g., Domeshek, 1992).
- *Learning*: arguments that the system's knowledge can be acquired from experience (or that its architecture can be selected for by evolution). A good example is Pazzani's (1988) account of the complementary role of similarity-based, theory-based, and explanation-based learning in the acquisition of causal knowledge.
- *Hardware-based*: certain hardware supports a given algorithm well. For example, the inference mechanisms of the NETL system for knowledge representation (Fahlman, 1979) are justified on the basis of efficient implementation on SIMD (single instruction, multiple data) hardware.

Whatever kind of explanation is employed, each of the design decisions in the descriptive part must be justified, to the extent that this is possible.

To sum up: in doing AI, it is not sufficient to describe a computer program; one must also specify the computational model that the program exemplifies, together with the range of tasks, domains, and environments that the model is intended to handle (the descriptive part of the AI theory). One must also provide functional justifications or teleological bases for the design of the computational model (the explanatory part of the AI theory). In addition, one must propose scientific hypotheses that makes a theoretical advance or contribution to the field. Finally, one must explain how the program and computational model supports (or disconfirms!) the hypotheses.

Conclusion

Foundations of Artificial Intelligence does not present a coherent set of "foundations of AI": instead, it presents a diverse collection of opinions regarding different aspects of the discipline. As a consequence, the book would benefit greatly from an integrative chapter, introduction, summary, or conclusion.

The book is definitely for the AI specialist; it is perhaps better suited for a special issue of a technical journal (as it was originally printed) than as a book. Although mostly well-written, the chapters are fairly hard to follow unless one already knows the issues pretty deeply. This does have its advantages, however: as a result, the book provides food for thought even for an expert in the field. The chapters are interesting and represent many of the major approaches to AI. Perhaps the best parts of the book are the critiques, since they are often from authors in different "camps," and they highlight many of issues that we have considered here.

In this review, we have attempted to provide a metatheoretic framework in which to integrate the many parallel efforts in the AI research arena; we hope that our framework will also serve as a methodological framework that can help drive research. With this framework in place, it is possible to see that much of the apparent disagreement between the authors of *Foundations of Artificial Intelligence* is one of emphasis rather than essential divergence of opinion. Different schools of thought in AI focus on different aspects of computation—architecture versus representation, for example—and advocate different styles of explanation and theory evaluation. However, it seems to us that the various approaches represented in the book are on the whole complementary, and provide valuable insights into different aspects of the nature of intelligence. If different research endeavors are pursued in full awareness of related alternatives, we believe that they can usefully constrain and inform one another.

Acknowledgements

Thanks to Peter Andreae for useful comments on a draft of this review.

References

Allen, J.F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

Agre, P.E. & Chapman, D. (1987). Pengi: An Implementation of a Theory of Activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 268–272.

Boden, M.A. (1977). *Artificial Intelligence and Natural Man.* Basic Books, New York.

Charniak, E. & McDermott, D. (1985). *Introduction to Artificial Intelligence.* Addison-Wesley Publishing Company, Reading, MA.

Cohen, P. (1991). A Survey of the Eighth National Conference on Artificial Intelligence: Pulling Together or Pulling Apart? *AI Magazine*, 12(1):16–41.

Davis, E. (1990). *Representations of Commonsense Knowledge.* Morgan Kaufmann Publishers, San Mateo, CA.

Dennett, D.C. (1978). *Brainstorms.* MIT Press, Cambridge, MA.

Domeshek, E.A. (1992). Do the Right Thing: A Component Theory for Indexing Stories as Social Advice, Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT.

Fahlman, S.E. (1979). *NETL: A System for Representing and Using Real-World Knowledge.* MIT Press, Cambridge, MA.

Hayes, P.J. (1985). The Second Naive Physics Manifesto. In J.R. Hobbs & R.C. Moore, editors, *Formal Theories of the Commonsense World*, 1–36, Ablex Publishing Corporation, Norwood, NJ.

Pazzani, M.J. (1988). *Learning Causal Relationships: An Integration of Empirical and Explanation-Based Learning Methods*, Ph.D. thesis, Technical Report UCLA-AI-88-10, University of California, Los Angeles, CA.

Schank, R.C. (1987). What is AI, Anyway? *AI Magazine*, 8(4):59–65.