To appear in
Artificial Intelligence.
Elsevier.

BOOK REVIEW

Building Large Knowledge-Based Systems: Representation and Inference in the CYC *Project*

Charles Elkan*
Department of Computer Science
University of California, San Diego
La Jolla, California 92093
elkan@cs.ucsd.edu

Russell Greiner†
Siemens Corporate Research
Princeton, New Jersey 08540
greiner@learning.siemens.com

The book under review here, *Building Large Knowledge-Based Systems: Representation and Inference in the* CYC *Project*, describes progress so far in an attempt to build a system that is intended to exhibit general common-sense reasoning ability. This review first discusses aspects of the CYC system, with a focus on important decisions made in designing its knowledge representation language, and on how claims about the performance of the system might be validated.[1] The review then turns to the book itself, discussing both its merits and its faults.

1 Comments on the CYC project

While today's expert systems can often perform their assigned tasks adequately, they are all narrow and "brittle": each is capable of dealing with only one small class of problems, and can reason sensibly only about the situations anticipated by its designers. These can be serious limitations. For example, a medical expert system will allow the age of a patient to be negative, unless its designer has specifically anticipated this problem and included in its knowledge base a special fact stating that this is impossible. Moreover, these narrow systems cannot readily extend what they already know, as they lack the analogical reasoning facility that would allow them, for example, to use information about treating one disease to help determine how to treat another.

The CYC project, under the leadership of Douglas Lenat and R. V. Guha, is an attempt to build an AI system that is neither narrow nor brittle. Lenat and Guha believe that the key to

*Supported by a grant from the Powell foundation and by the National Science Foundation under Award No. IRI-9110813. Both authors thank Devika Subramanian, Jan Wiebe and the reviewers for their comments on drafts of this review. The authors are listed alphabetically.

†Much of this work was performed at the Department of Computer Science of the University of Toronto, where it was supported by the Institute for Robotics and Intelligent Systems and by an operating grant from Canada's Natural Science and Engineering Research Council.

[1]The discussion is based on the description appearing in the book; the project has evolved significantly since the book was completed in mid-1989.

dealing with unanticipated situations as successfully as humans is to possess factual information and reasoning facilities as comprehensive as those possessed by humans. The CYC project therefore addresses the tremendous task of codifying a vast quantity of knowledge about the world, what the authors call "consensus reality" (page 4). The CYC team is attempting to capture the background knowledge possessed by a typical late twentieth century inhabitant of the United States. Lenat and Guha estimate that this will require at least ten million appropriately organized items of information, including rules and facts that describe concepts as abstract as causality and mass, as well as concepts as specific as the capital city of Texas.

The goals of the CYC project go beyond just making this mass of information available for retrieval. CYC therefore will include a wide range of reasoning facilities, including procedures for general deduction and analogical inference. These reasoning facilities are designed independently of the knowledge entered into CYC, since one hypothesis underlying the project is that knowledge can be entered without considering all of its future uses. CYC is intended to be able to give sensible answers to all sensible queries, not just those anticipated at the time of knowledge entry.

To date, the major effort in the project has been to determine exactly what information constitutes common-sense knowledge of the world, and how that knowledge should be represented and organized to allow its use in answering questions. Many other current research and development projects have a similar objective of finding appropriate formalizations of knowledge about basic aspects of the everyday world. Some of these are discussed below, but none is as comprehensive as CYC, and none devotes as much attention to merging results of work in different subfields into a coherent framework and a unified implementation, which is one of the major objectives of the CYC project.

As CYC is the first project of its magnitude, its designers have not been able to follow blueprints established by anyone else. Instead, they have made their own decisions about what categories of information should be included, how this information should be encoded and indexed, what inference mechanisms should be provided, how consistency should be maintained, and more. Our major concern with the project is the way in which these choices were made. While the authors do, on the whole, clearly describe the specific decisions they made, their descriptions of the other available options are often incomplete, and the arguments in favor of the chosen options are often unconvincing.

In some places, the authors mention cases where some other approaches do not work, as in the discussion of the nature of "things" (Section 4.1), of beliefs (Section 6.5.3), and of pieces of space (Section 6.7), but they do not argue for the generality of their approach. In other places, the authors simply assert that their chosen representation technique is better than certain others, for instance that it is more efficient to use typed quantifiers than unrestricted quantifiers (page 42). Occasionally the authors supply no arguments, but simply present the approach that they use, as for example their decision to focus on intervals of time, rather than points (Section 6.2.6).

It appears that in general the choices made by the designers of CYC were "non-decisions", that they felt it was more important to build something—anything—to see if it worked, than to contemplate at length which choices were in principle best. To a certain extent, this attitude of expediency is both admirable and necessary. Only an exertion of will can overcome doubt and allow progress in formalizing common-sense knowledge, since the final "truth" about concepts such as causation has not been discovered in thousands of years of thinking and will not be discovered by the Cyc team. Pragmatism allows the designers of Cyc to be satisfied with a partial formalization of causation and of other basic ideas, and to press on, rather than succumb to inaction.

Unfortunately, this attitude of expediency comes at the expense of carefully evaluating the factors and assumptions underlying Cyc design decisions. Discussions of these issues would not only satisfy academic kibitzers who wish to second-guess the Cyc choices, but also help users of Cyc, including new members of the Cyc team. In particular, knowing the assumptions underlying the design of the various parts of Cyc would allow potential users to predict better whether their intended uses of the system were likely to succeed.

Specific questionable decisions. The Cyc system is the product of numerous design decisions, ranging from the particular vocabulary it uses to a general perspective on how a common-sense reasoner should interface with users. The rest of this part of the review focuses on several of the debatable decisions made in the design of CycL, the knowledge representation language of Cyc. The claim here is not that these decisions are necessarily wrong, but that cogent arguments to defend the decisions would have been useful. We will discuss four important CycL design choices: the decisions that the language should be propositional, that frames should be the normal representation for propositions, that nonmonotonic inferences should be made only when explicitly sanctioned by the user, and that knowledge acquisition and inference should involve different languages between which translation is automatic.

The most fundamental aspect of CycL is that it is propositional: all knowledge is encoded in the form of logical sentences, rather than, for example, in diagrams, procedures, or numerical factors. The use of an endorsement-based scheme for managing uncertainty (Section 6.5.4) is one consequence of this decision for which the lack of a careful justification is especially noticeable. The authors do not sufficiently explain how the treatment of uncertainty as propositional in Cyc subsumes or improves on other implemented schemes for managing alternative assumptions and reasoning under uncertainty, notably reason maintenance systems [deKleer, 1986; Reiter and deKleer, 1987] and Bayesian networks [Pearl, 1988].

Different languages represent propositions in different ways. A second important choice in the design of CycL was the decision to make frames the primary way of representing propositions. While complex knowledge can be asserted in CycL as sentences in a logical language that includes the usual first-order predicate calculus connectives and quantifiers as well as certain special higher-order constructs, most information in CycL is stated and organized using frames. A simple fact, in particular, is asserted by storing a value in a slot of a frame. For example the fact that Austin is the capital of Texas is asserted by storing `Austin` as the value of the `Capital` slot of a frame for `Texas` (page 35).

The decision to use frames has several repercussions. First, it forces Cyc to maintain slots for inverses. When Austin is asserted to be the capital of Texas, for example, `Texas` must also be stored as the value of the `IsCapitalOf` slot of the frame for `Austin`, in addition to storing `Austin` as the value of the `Capital` slot of a frame for `Texas`. Stating facts involving relations of arity over two involves further complications (Section 3.2.3). The authors should explain why they rejected the alternative approach of using predicates and arguments. (Their claim that a frame-based encoding is more efficient (pages 36 and 63) is not well supported; see the next subsection.)

The slot/value approach to representing propositions has definite logical disadvantages. Storing `Texas` as the value of the `IsCapitalOf` slot of the frame for `Austin` amounts to treating `IsCapitalOf` as a predicate with subject `Austin` and object `Texas`. One of the most significant developments in logic in the 19th century was the jettisoning by Frege of the subject/object distinction in favor of a language where all arguments of a predicate are treated symmetrically [Frege, 1879]. This innovation made it possible to understand logical connectives and quantifiers independently, separating propositional logic from predicate logic. It also allowed the patterns of sound inference to be captured by a small number of rules instead of a large number of syllogisms.

A second repercussion of the CYCL commitment to frames is widespread reification, which can be expensive in both time and space. In CYCL, a new frame must be established for every aspect of an entity about which a fact is to be asserted (page 184). By contrast, with a predicate/argument formalism, aspects of an entity can be represented simply by compound terms. For example, to assert `Congestion( Capital(Texas), Moderate )` in a frame-based language, `Capital(Texas)` must first be reified into a new frame so that `Moderate` can be stored as the value of the `Congestion` slot of this frame. There are several reasons why using these frames can be significantly more expensive than simply using a compound term. For example, while a compound term such as `Mayor( Capital(Texas) )` intrinsically contains links to embedded subterms, such as `Capital(Texas)` here, the links between the frames reifying `Mayor( Capital(Texas) )` and `Capital(Texas)` must be asserted explicitly in a frame-based system.

The third important feature of CYCL is its treatment of nonmonotonic reasoning. Nonmonotonic rules permit conclusions that are not logically necessarily true, and so may require retraction later [Reiter, 1987]. A common example of a nonmonotonic rule is an assertion that the specified members of a set are its only members, which is called a closed world assumption. The CYC system can make inferences using nonmonotonic rules, but only if those inferences have been specifically sanctioned by a user, either by explicitly asserting an appropriate fact (for example that the closed world assumption is true of a specific set) or by explicitly invoking a nonmonotonic inference procedure.[2] This allows the user to change the set of conclusions that CYC can reach, perhaps based on the particulars of the domain and task.

In contrast to CYCL, many other knowledge representation languages use a fixed set of general syntactic principles to determine which nonmonotonic inferences should be sanctioned. For example, some languages apply the closed world assumption to the extension of all predicates, or use an ordering on predicates to determine which default rules preempt which others [Grosof, 1991]. The designers of CYC have recognized, at least implicitly, that it is wishful to hope that a single general set of criteria, independent of domain and task, can be sufficient to specify what nonmonotonic inferences are legitimate. They therefore allow users to specify what nonmonotonic inferences are

[2] In general, users can choose among many alternative inference procedures, where each procedure reflects a particular tradeoff between efficiency and completeness (page 128). Users can even choose procedures that may sacrifice soundness by, for example, making indiscriminate closed world assumptions.

admissible in a case-by-case way.[3]

A fourth interesting aspect of CYCL is the distinction between "epistemological" and "heuristic" levels of representation (Section 3.1.6). A user normally communicates with CYC in a high-level language called the epistemological level language. CYC translates queries and assertions in this language into a lower-level notation called the heuristic level language, which provides a variety of specialized inference mechanisms corresponding to special syntactic forms. The CYC system analyzes queries and assertions to decide which inference mechanisms are most appropriate, and encodes this choice in the heuristic level translation. For example, CYC can choose a path-following mechanism for queries it recognizes as taxonomic [Donini *et al.*, 1991].

The distinction between epistemological and heuristic levels has several benefits. One is that the designers of CYC can readily change the system to use new heuristic-level inference mechanisms, as they are implemented. Another benefit is that users can take advantage of the many special-purpose inference mechanisms in CYC, even if they themselves do not know which is appropriate when. However, the book does not clearly specify what information CYC uses when deciding which inference mechanism to use for a particular query.

It appears that CYC uses an opaque procedure that depends only on the syntactic form of a query, and that cannot be helped by hints provided by users. Other knowledge representation systems, for example MRS [Russell, 1985], represent control knowledge explicitly, in a form that allows both users and the system to employ standard reasoning methods to choose which inference mechanism is appropriate. This choice can be based on various types of information. One important type is estimates of the size of intermediate results; most database query optimization schemes are founded on this information. Another possible type is statistical information about the distribution of anticipated queries. Other systems use this to decide, for example, which subgoals to expand and in which order [Smith, 1989], or for which subqueries to memoize solutions [Chaudhri and Greiner, 1991].

Justifying claims of efficiency. The authors make the claim that various chosen inference mechanisms are efficient; see among other places, the general comments on pages 50 and 114 and the discussions of encoding schemes on pages 36 and 63, of typing information on page 43, of cost-effectiveness of multiple models on page 270, and of the `overrides` relation on page 304. Inferential efficiency is clearly vital if CYC is to remain useful as its knowledge base increases in size to capture more and more of common-sense knowledge. Measurements of efficiency are also vital as justifications for decisions by designers and users of CYC as to which of several reasoning techniques to use for a particular class of queries. Regrettably, the authors do not persuasively establish any claim of efficiency, as they do not provide any definition of efficiency, nor do they describe any method for demonstrating efficiency.

There are two standard approaches to demonstrating a performance claim. One approach is analytic: to prove that one technique is guaranteed to perform better than another under fixed

[3]Chapter 1 of the book discusses two so-called free lunch tries, each an attempt to obtain the benefits of possessing comprehensive common-sense knowledge while avoiding the laborious task of formalizing it explicitly. It would be fair to call the search for general principles of nonmonotonic reasoning, undertaken by other researchers, a third free lunch attempt.

conditions. Unfortunately, the literature in AI and in computer science in general provides few useful proof techniques. The great majority of analytical results deal with expressiveness or worst-case performance, so they produce results that are seldom helpful, as most languages for knowledge representation are equally expressive, and efficiency on typical inputs is usually more important than worst-case efficiency [Doyle and Patil, 1991].

A second approach to validation is statistical: to test experimentally whether one technique performs better than another over some sample population of inputs (assertions, retractions, queries, problems to solve, etc.). A well-designed experiment can produce statistically significant results under certain conditions, such as the sample being large enough and drawn from a stationary distribution, and so on.

The designers of CYC appear to be following a third, pragmatic, approach to validation. This approach is neither analytic nor experimental in the traditional statistical sense, since no explicit hypothesis is tested, and no random sample is taken. The CYC team considers an inference mechanism to be acceptably efficient if it provides tolerable response times for the queries on which it has been tested so far. *Ad hoc* experimentation of this sort can have numerous methodological pitfalls [Segre *et al.*, 1991b]. Moreover, even if one has reached valid conclusions about the performance of an inference mechanism on one population, these conclusions may not apply to a different population. For example, conclusions valid for small problems will only apply to large problems given certain assumptions concerning the internals of the inference mechanism used, assumptions that can be difficult to test [Segre *et al.*, 1991a].

This concern that conclusions that pertain to one population may be applied inappropriately to a different population is relevant to the CYC project, as there is a good reason to expect that tomorrow's applications of CYC will not resemble today's. Successful software systems tend to be used as back-ends in later larger systems. When a system is used as a back-end, often new front-ends are introduced that generate inputs that no longer resemble any natural human input. Unfortunately, even if the performance of a system is adequate on natural queries, it may well be inadequate on syntactically similar but unnatural queries. The history of the Unix EGREP utility is an example of this phenomenon.[4] A similar evolution can be expected to occur with CYC, especially since it is intended to be used as a back-end to overcome the brittleness of specialized expert systems.

Evaluating the progress of the CYC project. The criterion advanced by the designers of CYC for judging the overall success and progress of the project is as informal as their criterion for evaluating the efficiency of the system. Success is defined as the system "really working", that is, being used by many people for further research and in developing new production expert systems. The book, however, does not discuss how to estimate whether the project is on course towards this goal.

---

[4] `egrep` is a tool for matching regular expression patterns. As first implemented by Al Aho [personal communication, 1991], it translated regular expressions directly into finite state machines. However, a new front-end (a calendar utility with a very flexible language for describing dates and ranges of dates) later sent extremely complex expressions as input to `egrep`. On these expressions the original version of `egrep` could require 30 seconds to construct a finite state machine that then required 0.3 seconds to process its input. Aho therefore reimplemented `egrep` to use a lazy algorithm for constructing state transitions, at the cost of a slowdown of less than 15% in following transitions.

One way of addressing this issue would be a longitudinal study, using empirical measurements taken at different times to determine whether the emerging CYC system is improving. An important type of improvement that should be observable is a decrease in the effort required to enter a piece of information, as more and more reuse of existing knowledge becomes possible. The coverage and accuracy of CYC could also be measured quantitatively [Greiner and Elkan, 1991]: over a wide population of factual queries developed independently, how often does CYC provide answers, and how often are these answers correct? The efficiency with which CYC produces these answers should also be measured over time, to determine whether more knowledge allows queries to be answered faster, as hoped. For other systems this is not always the case: the notorious utility problem is that incorporating additional learned rules can actually lower the overall performance of a problem solver [Minton, 1990].

The questions above all ask for real studies and measurements of the performance of the CYC system as a whole. The same empirical methodology should be used to investigate the utility of specific components of CYC: whether the system is more efficient, more accurate, etc. with one of its reasoning methods rather than without. The utility of general facts and rules should also be tested: how the coverage, accuracy or efficiency of CYC changes if we remove, for example, a specific nonmonotonic rule, such as one that sanctions indiscriminate closed world assumptions.

Quantitative empirical information may be useful to the CYC designers as they consider which new facilities to incorporate in the system. For example, the authors argue convincingly that analogical reasoning provides a very important way of overcoming brittleness (Section 1.4) and so they plan to provide the CYC system with facilities for proposing appropriate analogies and then using them effectively.[5] They claim that there may be many different species of analogical reasoning (page 14). Experiments should be run to determine which of these types of analogical reasoning more often find analogies, and which obtain more useful analogies. A related empirical question is whether CYC will be able to find and exploit better analogies as its knowledge base increases in size, or whether analogical reasoning falls victim to a variant of the utility problem.

So-called Socratic questioning can be used to make people aware of knowledge that they possess implicitly but initially cannot articulate. While the CYC system, unlike humans, can always access all of its knowledge, a similar questioning protocol may still be valuable as a focusing mechanism. The ONTIC system [McAllester, 1989] can be guided to discoveries by asking it to focus on particular objects in its knowledge base, and so-called "access-limited logic" involves a similar idea [Crawford and Kuipers, 1991]. Similar techniques may be important with CYC. Once again, the contribution of this type of reasoning to CYC should be investigated experimentally.

2 Comments on the book itself

The book under review here is quite different in style from most academic books. On the positive side, the writing is generally clear and there are many witty comments, as one would expect from its authors. The book does an excellent job of motivating the overall project, and then discusses,

[5]Unfortunately, they have postponed serious work on analogy in favor of more intensive knowledge entry.

in some detail, how its authors, together with other members of the Cyc community, are actually building the Cyc system. It provides many helpful examples, and often discusses ideas that were considered but abandoned, for example when covering belief (Section 6.5.3.1), certainty factors (Section 6.5.4.1) and space (Section 6.7.1).

On the negative side, the book does not conform to the usual standards of scientific writing that apply to technical monographs. As its authors want and deserve a wide readership, it is understandable that they worked to avoid pedantry. However, it is precisely the fact that the book has had and will have many readers who are not experts on knowledge representation that makes it regrettable that it does not follow all professional conventions. The book deviates from convention in two ways.

The less important deviation is the highly colloquial language of the book. Conversationalism in the pursuit of understandability is no sin, and the book is remarkably free of jargon. However, colloquialism becomes problematic when the authors fail to distinguish between statements that are meant to be taken literally, and statements that are intended to be only suggestive. As one specific example, readers may be unable to determine whether the scenarios discussed in Section 3.3.21 have really been implemented. Sometimes colloquialism shades into sloppiness. There are a number of typographical errors (some are quite amusing, as "modus *powers*" on page 49) and several terms are used before they are defined, while others are defined more than once. In addition, several references are cited but do not appear in the bibliography, for example "[McCarthy81]" on page 39, and there are many references to the literature and implemented systems with no actual citations, for example to Hayes and McCarthy on page 81 and to Schank on page 223.

This leads to the second, more important deviation from convention: the paucity of bibliographic citations in the book. It is difficult (or sad) to believe that the entire background for the Cyc project can be captured in only 22 publications, of which 7 were written by Lenat. While the Cyc project is the only attempt to formalize all aspects of common sense knowledge in a coherent way, numerous other research and development projects are also concerned with formalizing basic everyday knowledge. To name just a few subfields, there are many projects that are actively seeking useful theories of belief systems, uncertainty management, temporal reasoning, causality, and meta-level reasoning.[6] Given that research in many of these knowledge representation areas is pre-paradigmatic (that is, no single theory or model is widely accepted as a basis for further investigation [Kuhn, 1970]), the discussions of Cyc design decisions, especially, should include more extensive comparisons with related work. Unfortunately, the book does not systematically discuss related work by others, nor does it even include many of the pointers needed to allow a reader to investigate alternative schemes and opposing arguments. These omissions makes the book significantly less useful than it could be, especially to anyone who is not already an expert in AI.

The lack of references could furthermore mislead some readers into overestimating the degree of originality of the Cyc work. The project stands out in comparison to almost all other research

[6]This review has provided a few specific pointers to relevant work above. For reprints of many important papers, see the Morgan Kaufman series of anthologies, including *Readings in Knowledge Representation* [Brachman and Levesque, 1985], *Readings in Nonmonotonic Reasoning* [Ginsberg, 1987], *Readings in Planning* [Allen *et al.*, 1990] and *Readings in Uncertain Reasoning* [Shafer and Pearl, 1990].

in artificial intelligence not for the power of its new ideas, but for the power of its vision and for the fact that it actually sets out to implement what have often been thus far just theoretical ideas.

3 Conclusion

The book reviewed here provides many interesting insights obtained from the years that the authors have spent attempting to formalize common-sense knowledge. It has already found many readers, and should find more. However, these readers will not always find what they might legitimately expect.

Researchers in the area of knowledge representation will find that many important technical details of the design of the CYC system are missing, along with comparisons to similar work by others. The book can still be valuable for them, because it does bring up a host of the real-world issues that must be addressed by anyone who wants to build a practical representation and reasoning system. In an invited talk at the KR'89 conference, David Poole issued the battle cry "Get thee to a terminal!" The CYC team are heeding this injunction, and their practical experience furnishes the authors with many insights that other researchers should reflect upon.

Knowledge engineers in industry, on the other hand, who are developing expert systems with current technology, will find few insights in the book that are directly useful. Moreover, if CYC is successful, in that future expert systems tap into it, then this book will be of only historical importance: its successor (or perhaps great-grand-successor) will be more up-to-date, and hence more relevant. We do recommend the book, nevertheless, to anyone who plans to work with expert systems for a while, as it addresses many issues that will be important as systems begin to scale up, including how to structure very large knowledge bases and how to incorporate advice from many different experts.

Overall, despite the reservations that constitute the major part of this review, we have high long-term expectations for the CYC project. The project is still in its early stages (the book was written when it was five years into an initialization phase scheduled to last ten years), and we look forward to reading the already-cited 1997 description of the more nearly finished system. We hope that the next CYC book will include significant demonstrations of the successful ideas, and thorough discussions of why the unsuccessful ideas failed.

References

[Allen *et al.*, 1990] James Allen, James Hendler, and Austin Tate, editors. *Readings in Planning.* Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1990.

[Brachman and Levesque, 1985] Ron Brachman and Hector Levesque, editors. *Readings in Knowledge Representation.* Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1985.

[Chaudhri and Greiner, 1991] Vinay Chaudhri and Russell Greiner. A formal analysis of solution caching. Technical report, University of Toronto, 1991.

[Crawford and Kuipers, 1991] J.M. Crawford and B.J. Kuipers. Negation and proof by contradiction in access-limited logic. In *AAAI-91*, pages 897–903, Anaheim, July 1991.

[deKleer, 1986] Johan deKleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28(2):127–162, 1986.

[Donini *et al.*, 1991] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In *Proceeding of 1991 Conference on Knowledge Representation*, pages 151–62, Boston, April 1991.

[Doyle and Patil, 1991] Jon Doyle and Ramesh Patil. Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48(3), 1991.

[Frege, 1879] Gottlob Frege. *Begriffsschrift*, a formula language, modeled upon that of arithmetic, for pure thought, 1879. Reprinted in [van Heijenoort, 1967].

[Ginsberg, 1987] Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987.

[Greiner and Elkan, 1991] Russell Greiner and Charles Elkan. Measuring and improving the effectiveness of representations. In *IJCAI-91*, pages 518–24, Sydney, Australia, August 1991.

[Grosof, 1991] Benjamin Grosof. Generalizing prioritization. In *Proceeding of 1991 Conference on Knowledge Representation*, pages 289–300, Boston, April 1991.

[Kuhn, 1970] Thomas S. Kuhn. *The Structure of Scientific Revolution.* University of Chicago Press, Chicago, Ill., 1970. Enlarged edition.

[McAllester, 1989] David A. McAllester. *Ontic : A Knowledge Representation System for Mathematics.* MIT Press, Cambridge, Mass., 1989.

[Minton, 1990] Steven Minton. Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence*, 42(2–3):363–391, March 1990.

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

[Reiter and deKleer, 1987] Raymond Reiter and Johan deKleer. Foundations of assumption-based truth maintenance systems. In *AAAI-87*, pages 183–88, Seattle, Washington, July 1987.

[Reiter, 1987] Raymond Reiter. Nonmonotonic reasoning. In *Annual Review of Computing Sciences*, volume 2, pages 147–87. Annual Reviews Incorporated, Palo Alto, 1987.

[Russell, 1985] Stuart Russell. *The Compleat Guide to MRS*, June 1985. Stanford KSL Report HPP-85-12.

[Segre *et al.*, 1991a] Alberto Segre, Charles Elkan, Geoffrey Gordon, and Alexander Russell. A robust methodology for experimental evaluations of speedup learning, June 1991. Working paper available from the authors.

[Segre *et al.*, 1991b] Alberto Segre, Charles P. Elkan, and Alex Russell. A critical look at experimental evaluations of EBL. *Machine Learning Journal*, 6(2), 1991.

[Shafer and Pearl, 1990] Glenn Shafer and Judea Pearl, editors. *Readings in Uncertain Reasoning.* Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1990.

[Smith, 1989] David E. Smith. Controlling backward inference. *Artificial Intelligence*, 39(2):145–208, June 1989.

[van Heijenoort, 1967] Jean van Heijenoort, editor. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931.* Harvard University Press, 1967.