The Knowledge Engineering Review, Vol. 20:4, 431–433.
Printed in the United Kingdom

Book Reviews

Knowledge in action: Logical foundations for specifying and implementing dynamical systems
by Raymond Reiter, MIT Press, 0-262-18218-1, 448 pp., $60.00/£38.95
doi:10.1017/S0269888906210749

When I started out as a newly hatched PhD student, back in the day, one of the first articles I read and understood (or at least thought that I understood) was Ray Reiter's classic article on default logic (Reiter, 1980). This was some years after the famous 'non-monotonic logic' issue of *Artificial Intelligence* in which that article appeared, but default logic was still one of the leading approaches, a tribute to the simplicity and power of the theory. As a result of reading the article, I became fascinated by both default logic and, more generally, non-monotonic logics. However, despite my fascination, these approaches never seemed terribly useful for the kinds of problem that I was supposed to be studying—problems like those in medical decision making—and so I eventually lost interest. In fact non-monotonic logics seemed to me, and to many people at the time I think, not to be terribly useful for anything. They were interesting, and clearly relevant to the long-term goals of Artificial Intelligence as a discipline, but not of any immediate practical importance.

This verdict, delivered at the end of the 1980s, continued, I think, to be true for the next few years while researchers working in non-monotonic logics studied problems that to outsiders seemed to be ever more obscure. However, by the end of the 1990s, it was becoming clear, even to folk as short-sighted as I, that non-monotonic logics were getting to the point at which they could be used to solve practical problems. *Knowledge in action* shows quite how far these techniques have come.

The reason that non-monotonic logics were invented was, of course, in order to use logic to reason about the world. Our knowledge of the world is typically incomplete, and so, in order to reason about it, one has to make assumptions about things one does not know. This, in turn, requires mechanisms for both making assumptions and then retracting them if and when they turn out not to be true. Non-monotonic logics are intended to handle this kind of assumption making and retracting, providing a mechanism that has the clean semantics of logic, but which has a non-monotonic set of conclusions.

Much of the early work on non-monotonic logics was concerned with *theoretical* reasoning, that is reasoning about the beliefs of an agent—what the agent believes to be true. Theoretical reasoning is the domain of all those famous examples like 'Typically birds fly. Tweety is a bird, so does Tweety fly?', and the fact that so much of non-monotonic reasoning seemed to focus on theoretical reasoning was why I lost interest in it. I became much more concerned with *practical* reasoning—that is reasoning about what an agent should do—and non-monotonic reasoning seemed to me to have nothing interesting to say about practical reasoning. Of course I was wrong. When one tries to formulate any kind of description of the world as the basis for planning, one immediately runs into applications of non-monotonic logics, for example in keeping track of the state of a changing world.

It is this use of non-monotonic logic that is at the heart of *Knowledge in action.* Building on the McCarthy's situation calculus, *Knowledge in action* constructs a theory of action that encompasses a very large part of what an agent requires to reason about the world. As Reiter says in the final chapter,

> we have developed situation calculus accounts for time, processes, concurrency, procedures, exogenous events, reactivity, sensing and knowledge, probabilistic uncertainty, and decision theory.

That is an impressive list, and an agent equipped with a theory that handles all of those aspects of the world is able to cope with a large part of the world's complexity. However, the development of this theory is not all that the book contains. While the theory was being developed, Reiter's group was also developing Golog, a logic programming language, implemented on top of Prolog, that allows one to write programs that reason using the theory (and the code for the Golog interpreter is distributed by the group).

What this means is that not only does *Knowledge in action* make it possible to write down, essentially in logic, complex descriptions of the world—in other words descriptions that deal explicitly with much of the complexity of the world that one has to take into account in order to get around the world effectively—and construct plans of action from them, but it is also possible to construct these plans automatically, and to execute them. I find this very impressive, and vindication, in many ways, of the entire program that saw the route to Artificial Intelligence as being through the formalization of information about the world in logic and the running of theorem provers on the results of that formalization.

Particularly interesting to me is the way that, in the final chapters, Reiter shows how to combine the logic-based view of the world that he has developed up to that point with the use of probability and decision theories. The latter are essential in handling the stochastic nature of our knowledge about the world[1], and combining them effectively with logic has been another open challenge.

As a result, *Knowledge in action* is, I think, something of a landmark. With both theory and implementation complete, it provides an off-the-shelf mechanism that can be plugged into an agent to enable it to figure out what to do, something that has not been previously available. This is a mechanism that is crying out to be used, and I will certainly be pushing my graduate students to use it.

Knowledge in action is not a newly published book. It came out in October 2001, just over 4 years before the date of writing this review. The reason for the delay in my writing is not important, but it does mean, sadly, that I never got to tell the author, however indirectly, how much I liked his book. He died, at the age of 63, about a year after the book appeared. I cannot say I knew him—I met him a couple of times in that awkward way that one meets senior people in the field at conferences—but from what I saw, some of his spirit lives on in this book. Careful, scholarly, quietly spoken, and one that many of us will look up to for a long while yet.

Reviewed by Simon Parsons
Brooklyn College, City University of New York, USA

References

Reiter, R, 1980, A logic for default reasoning. *Artificial Intelligence* **13**, 81–132.

Introduction to Machine Learning by Ethem Alpaydin, MIT Press, 0-262-01211-1, 400 pp., $50.00/£32.95
doi:10.1017/S0269888906220745

These days, in most computer science departments, undergraduates have the chance to take more than one course in Artificial Intelligence (AI). The course that used to be *the* course on AI, the introductory course that covers logic and search and other basic topics, is now often given earlier in the curriculum—in the penultimate year in many departments—allowing interested students to delve deeper into more advanced aspects of the subject in additional courses. One area that might

[1] That is a bit of a gloss. I do not think we need to use probability and decision theories *per se*, but we do need to be able to do the kinds of things that one can do with them—that is we need mechanisms for representing and reasoning with uncertain information and making decisions on the basis of uncertain information.

profitably be covered in one of these additional courses is machine learning, an area that is maturing to the point that it is becoming widely applied in mainstream IT. Indeed, machine learning has now grown to a point that it itself needs an introductory course devoted to it, and there are several reasons why *Introduction to Machine Learning* would be a good textbook for such a course.

For a start, *Introduction to Machine Learning* covers a lot of ground. After a suitable introduction—to the general idea of machine learning, some of the basic concepts that recur in the rest of the book, and Bayesian probability and decision theory—the book contains separate chapters on parametric methods, multivariate methods, dimensionality reduction, clustering, non-parametric methods, decision trees, linear discrimination, multilayer perceptions, local models, hidden Markov models, assessing and comparing classification algorithms, combining multiple learners, and, finally, reinforcement learning. To me that seems to be plenty of material for a one semester course. Indeed, it is probably much more material than one could possibly get through in such a course, at least without overwhelming the students.

The treatment of the material seems to me to be at about the right level of detail as well. Of course, the book does not cover all of the topics it discusses in anything like full detail. Indeed, it does not really discuss any of the topics that it covers in anything like full detail. However, this is perfectly reasonable, not least because each could be the subject of an entire book itself. What *Introduction to Machine Learning* does is to give enough detail that the reader comes away with the basic concepts, and the book provides enough pointers that the reader who wants to know more can get to that extra material quickly. This treatment is exactly what I expect this kind of good introductory textbook to do. It is not meant to be a textbook that supports a course on any of the individual topics it covers—there are other books to do that—it is only intended to support a course on *all* of the topics, and to me it seems to do that very neatly.

One could also, of course, pick holes in the coverage, extensive though it is. The book scarcely mentions association rules, for example, giving them exactly one page (page 56), at least as far as I could see. If you happen to like association rules you might feel that this somewhat under-represents their place in the machine learning literature. However, paring down the whole of machine learning into a single course involves taking some things out, and overall I do not feel that *Introduction to Machine Learning* does any sub-area of machine learning too great a disservice. Indeed, by providing a clear and careful introduction to so many sub-areas, and thus increasing the number of people who are exposed to it—some of whom will, hopefully, be interested enough that they eventually take courses that explore some of the topics in more detail—I think the book makes a great contribution to the machine learning literature.

Reviewed by Simon Parsons
Brooklyn College, City University of New York, USA