Finally, the book presents a number of articles addressing the issues that are closely related to planning, for example, plan control, plan execution, resource allocation and scheduling.

The book is a collection of research outcomes that present the significant aspects of progress in current AI planning. The emphasis is upon various logical formalisms and reasoning mechanisms to support deductive planning, and some of the approaches are certainly novel and promising. Another feature of the book is that many of the authors carry out their discussions in a very formal, concise way, such that main theoretical and technical issues in the articles are accurately addressed and expressed. As a workshop proceedings, however, the book can only portray the special interests of those who attended. Although deductive planning is a central subject of the book, the papers do not provide a very representative cross-section of work in the domain. With the introduction of a variety of non-classical logics, for example, a central issue for deductive planning is to develop sound proof theories supporting these reasoning patterns for plan generation. However, most of the articles in the book emphasize logical formalisms for plan representation without providing sufficient discussions of deductive reasoning procedures for plan generation. Besides, none of the papers addresses non-monotonic reasoning which is one of the most important logical formalizations in AI planning. Finally, it is my view that the book covers too few aspects of AI planning to match the title.

The book is not organized into subject-based sections. Nor does it have an index. It is not, therefore, a book for teachers or undergraduate students, nor indeed for practitioners at the practical application end. But for the researchers in AI planning, in automated reasoning, or in knowledge processing, the book is well worth reading.

Reviewed by Dr Huaming Lee and Mr Jon Sims Williams, Department of Engineering Math, University of Bristol, Bristol BS8 1TR, UK

Logic programming: operational semantics and proof theory by J. H. Andrews, Cambridge University Press, 1992, pp 104, £25.00, ISBN 0-521-43219-7.

This book is published by Cambridge University Press in its distinguished dissertation series in computer science. It does not qualify as a textbook, but researchers interested in the semantics of both sequential and parallel logic programming will find it an important contribution to the field.

The thesis provides operational semantics and proof theoretic characterizations for systems having each of the possible combinations of parallel or sequential "and" and parallel or sequential "or". A proof theoretic characterization associates a program with a proof system with respect to which the operational semantics is sound and complete.

The proposed operational semantics, Stack-of-Stacks (SOS), operates on programs consisting of definitions of predicates by goal-formula bodies, as opposed to SLD-resolution which operates on programs consisting of Horn clauses. Thus the proposed operational semantics manipulates goal-formulae as its basic units. The rules for SOS are simple and very predictable. The variants of SOS correspond to the parallel "and" and "or", sequential "and", sequential "or", and sequential "and" and "or" control disciplines.

The proof theoretic characterization is in the form of sequent calculi. The core part, LKE, of this calculi is basically a Gentzen-style sequent calculus for first-order logic with equality. LKE augmented with a set of axioms describes the success and failure behaviour of queries under certain assumptions. For example, LKE augmented with a set of PAR axioms characterizes the queries which succeed in parallel-or systems and those which fail in parallel-and systems. This means that if a query A succeeds (resp. fails) in a parallel-or (resp. parallel-and) system, then the sequent $[\rightarrow S(\exists[A])]$ resp. $[\rightarrow F(\exists[A])]$ is derivable from LKE + PAR. Similarly, LKE + SEQ characterizes those queries which succeed in the sequential-and, sequential-or system, and those which fail in sequential-and system; LKE + PASO characterizes those queries which succeed in the parallel-and, sequential-or system.

The book contains definitions and theorems on almost every page, and many of them are generalizations of results related to Horn clause logic programming. Reinterpretation of some well known results of mathematical logic in this domain is quite interesting, for example, the Church–Rosser Property of SOS and the Halting Problem. Although the author provides a very fluid introduction, the rest of the material is very theoretical and difficult to follow, due to the lack of enough motivating examples.

Reviewed by Subrata K. Das, Queen Mary and Westfield College, Mile End Road, London E1, UK

Formal specification of complex reasoning systems edited by Jan Treur and Thomas Wetter, Ellis Horwood, Chichester, 1993, pp 282, £35.95, ISBN 0-13-336-785-1.

In its youth, KBS research was vigorous and dynamic, but with a disturbing tendency to impulsiveness and ill-considered assertion. With the onset of middle-age, the confidence of adolescence has given way to a more introspective mood, with an accompanying desire to be taken seriously by the wider research community. Two of the most obvious role models are software engineering (with its emphasis for standardized methodologies and tools for program development) and computational logic (which supplies methods for describing and proving properties of systems). This book's primary theme is the use of computational logic for the specification of KBS, but this is intertwined with a methodological subtheme which subtly influences the authors' choices of formal systems. The general approach is to provide a snapshot of recent research across a variety of research groups. Eight different systems are described and, to provide a platform for comparison, a standard example is used by each system. Although the example is (of necessity) quite small, it provides an invaluable anchoring point for the more abstract technical discussions.

Although the book is organized as a series of collected papers, sandwiched between an introduction to the running example and a conclusion which draws comparisons between the systems, it is more interesting to review it by drawing out some of the dominant issues which surface sporadically throughout. The most noticeable of these is the choice of specification language, which varies considerably between systems. It is ironic (given that these languages are intended to rationalize KBS design) that it is difficult to provide a direct comparison of them, although a brave attempt is made in the concluding chapter. This is an example of the subtle influence of methodology on formal methods, since the differing "views of the world" taken by each of the research groups has clearly exerted a strong influence on the shape of their formal systems. This divergence is (I suspect) further emphasized by the high level of expertise of each group, which makes it possible for them to select the formal framework which is "just right" for their particular approach. As a result, we are presented with a range of specialized logical frameworks, tuned to each research group's interests. Although some of these sit directly on top of established specification languages (such as OBJ3), most are blends of different varieties of computational logic, applied to different stages of KBS development—for example, $K_{BS}SF$ uses both an algebraic specification language and first order predicate calculus. Many of the systems described are designed to accommodate this diversity using various types of modularity—with refinement between modules in MILORD; design by composition of modules in DESIRE and communication by bridge rules between modular theories in MC.

Along with the emphasis for modularity comes the desire to stratify the specification task in some way. Earlier work on "semi-formal" KBS design methodologies (KADS in particular) has exerted a strong influence on the solutions proposed here. ML^2 explicitly takes KADS as its starting point and offers formal accounts of the domain, inference and task levels of development—the general idea being that the inference level transfers information to and from the domain level whilst the task level overlays control information on the inference level. A similar approach is taken in the KARL system, although using a more object-oriented style of specification and with more weight given to the visualization of each level using diagrammatic notations. Both ML^2 and