Modal Logic

Patrick Blackburn, Maarten de Rijke
and Yde Venema

CAMBRIDGE TRACTS
IN THEORETICAL
COMPUTER SCIENCE
53

Modal Logic

Patrick Blackburn
LORIA, Nancy

Maarten de Rijke
University of Amsterdam

Yde Venema
University of Amsterdam

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011–4211, USA
10 Stamford Road, Oakleigh, VIC 3166, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2001

Printed in the United Kingdom at the University Press, Cambridge

Typeset by the author in Times and Computer Modern 10/13pt

A catalogue record of this book is available from the British Library

Library of Congress Cataloguing in Publication data

Blackburn, Patrick, 1959–
Modal logic / Patrick Blackburn, Maarten de Rijke, Yde Venema.
p. cm.
Includes bibliographical references and index.
ISBN 0 521 80200 8
1. Modality (Logic). I. Rijke, Maarten de. II. Venema, Yde, 1963– III. Title.
QA9.46.B58 2001
511.3 21–dc21 00-054667

ISBN 0 521 80200 8 hardback

MODAL LOGIC

Cambridge Tracts in Theoretical Computer Science

Titles in the series

1. G. Chaitin *Algorithmic Information Theory*
2. L. C. Paulson *Logic and Computation*
3. M. Spivey *Understanding Z*
5. A. Ramsey *Formal Methods in Artificial Intelligence*
6. S. Vickers *Topology via Logic*
7. J.-Y. Girard, Y. Lafont & P. Taylor *Proofs and Types*
8. J. Clifford *Formal Semantics & Progmatics for Natural Language Processing*
9. M. Winslett *Updating Logical Databases*
10. K. McEvoy & J. V. Tucker (eds) *Theoretical Foundations of VLSI Design*
11. T. H. Tse *A Unifying Framework for Structured Analysis and Design Models*
12. G. Brewka *Nonmonotonic Reasoning*
14. S. G. Hoggar *Mathematics for Computer Graphics*
15. S. Dasgupta *Design Theory and Computer Science*
17. J. C. M. Baeten (ed) *Applications of Process Algebra*
18. J. C. M. Baeten & W. P. Weijland *Process Algebra*
19. M. Manzano *Extensions of First Order Logic*
21. D. A. Wolfram *The Clausal Theory of Types*
22. V. Stoltenberg-Hansen, I. Lindström & E. Griffor *Mathematical Theory of Domains*
23. E.-R. Olderog *Nets, Terms and Formulas*
26. P. D. Mosses *Action Semantics*
27. W. H. Hesselink *Programs, Recursion and Unbounded Choice*
28. P. Padawitz *Deductive and Declarative Programming*
29. P. Gärdenfors (ed) *Belief Revision*
30. M. Anthony & N. Biggs *Computational Learning Theory*
31. T. F. Melham *Higher Order Logic and Hardware Verification*
32. R. L. Carpenter *The Logic of Typed Feature Structures*
33. E. G. Manes *Predicate Transformer Semantics*
34. F. Nielson & H. R. Nielson *Two Level Functional Languages*
35. L. Feijs & H. Jonkers *Formal Specification and Design*
36. S. Mauw & G. J. Veltink (eds) *Algebraic Specification of Communication Protocols*
37. V. Stavridou *Formal Methods in Circuit Design*
38. N. Shankar *Metamathematics, Machines and Gödel's Proof*
39. J. B. Paris *The Uncertain Reasoner's Companion*
40. J. Dessel & J. Esparza *Free Choice Petri Nets*
41. J.-J. Ch. Meyer & W. van der Hoek *Epistemic Logic for AI and Computer Science*
42. J. R. Hindley *Basic Simple Type Theory*
43. A. Troelstra & H. Schwichtenberg *Basic Proof Theory*
44. J. Barwise & J. Seligman *Information Flow*
45. A. Asperti & S. Guerrini *The Optimal Implementation of Functional Programming Languages*
46. R. M. Amadio & P.-L. Curien *Domains and Lambda-Calculi*
47. W.-P. de Roever & K. Engelhardt *Data Refinement*
48. H. Kleine Büning & T. Lettman *Propositional Logic*
49. L. Novak & A. Gibbons *Hybrid Graph Theory and Network Analysis*
51. H. Simmons *Derivation and Computation*
52. A. S. Troelstra & H. Schwictenberg *Basic Proof Theory* (Second Edition)
53. P. Blackburn, M. de Rijke & Y. Venema *Modal Logic*

For Johan

Preface

Ask three modal logicians what modal logic is, and you are likely to get at least three different answers. The authors of this book are no exception, so we will not try to start off with a neat definition. Nonetheless, a number of general ideas guide our thinking about the subject, and we will present the most important right away as a series of three slogans. These are meant to be read now, and, perhaps more importantly, referred back to occasionally; doing so will help you obtain a firm grasp of the ideas and intuitions that have shaped this book. Following the slogans we will discuss the aims and content of the book in more detail.

Our first slogan is the simplest and most fundamental. It sets the basic theme on which the others elaborate:

Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures.

In this book we will be examining various *propositional* modal languages: that is, the familiar language of propositional logic augmented by a collection of *modal operators*. Like the familiar boolean connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\bot$, and $\top$), modal operators do *not* bind variables. Thus, as far as syntax is concerned, we will be working with the simplest non-trivial languages imaginable.

But in spite of their simplicity, propositional modal languages turn out to be an excellent way of talking about *relational structures*, and this book is essentially an attempt to map out some of the ramifications of this. For a start, it goes a long way towards explaining the recent popularity of modal languages in applied logic. Moreover, it introduces one of the fundamental themes in the mathematical study of modal logic: the use of relational structures (that is, *relational semantics*, or *Kripke semantics*) to explicate the logical structure of modal systems.

A *relational structure* is simply a set together with a collection of relations on that set. Given the broad nature of this definition, it is unsurprising that relational structures are to be found just about everywhere. Virtually all familiar mathematical structures can be thought of as relational structures. Moreover, the entities commonly used to model the phenomena of interest in various applications often turn out to be relational structures. For example, theoretical computer scientists use labeled transition systems to model program execution, but a labeled transition system is just a set (the states) together with a collection of binary relations (the transition relations) that model the behavior of programs. Moreover, relational structures play a fundamental modeling role in many other disciplines, including knowledge representation, computational linguistics, formal semantics, economics, and philosophy. As modal languages are the simplest languages in which relational structures can be described, constrained, and reasoned about, it is hardly surprising that applied modal logic has blossomed in recent years.

But relational structures have also played a fundamental role in the development of the mathematics of modal logic: their use turned modal logic from a rather esoteric branch of syntax manipulation into a concrete and intuitively compelling field. In fact, it is difficult to overstate the importance of relational models to modal logic: their discovery in the 1950s and early 1960s was the biggest single impetus to the development of the field. An early application was *completeness theory*, the classification of modal logics in relational terms. More recently, relational semantics has played an important role in mapping out the *computational complexity* of modal systems.

Modal languages may be simple – but what makes them special? Our next slogan tries to pin this down:

Slogan 2: Modal languages provide an internal, local perspective on relational structures.

That is, modal languages talk about relational structures in a special way: 'from the inside' and 'locally.' Rather than standing outside a relational structure and scanning the information it contains from some celestial vantage point, modal formulas are evaluated *inside* structures, *at a particular state*. The function of the modal operators is to permit the information stored at other states to be scanned – but, crucially, *only the states accessible from the current point via an appropriate transition may be accessed in this way*. This idea will be made precise in the following chapter when we define the satisfaction definition. In the meantime, the reader who pictures a modal formula as a little automaton standing at some state in a relational structure, and only permitted to explore the structure by making journeys to neighboring states, will have grasped one of the key intuitions of modal model theory.

The internal perspective modal languages offer makes them natural for many applications. For a start, the decidability of many important modal systems stems from the step-by-step way that modal formulas are evaluated. Moreover, in a number of disciplines, simple languages offering an internal perspective on relational structures have been devised; sometimes these (independently invented) systems turn out to be variants of well-known modal systems, and can be analyzed using modal techniques. For example, Kasper-Rounds logic (used in computational linguistics) is essentially a natural notation for a certain fragment of propositional dynamic logic with intersection, and many of the description logics used in knowledge representation can be usefully viewed as (fragments of) modal languages. Finally, it is also the stepwise way in which modal formulas are evaluated which explains why the notion of *bisimulation*, a crucial tool in the process theoretic study of labeled transition systems, unlocks the door to important characterizations of modal expressivity.

So far there have been only two characters in this discussion: modal languages and the structures which interpret them. Now it is certainly true that for much of its history modal logic was studied in isolation, but the true richness of the subject only becomes apparent when one adopts a broader perspective. Accordingly, the reader should bear in mind that:

Slogan 3: Modal languages are not isolated formal systems.

One of the key lessons to have emerged since about 1970 is that it is fruitful to systematically explore the way modal logic is related to other branches of mathematical logic. In the pair ⟨MODAL LANGUAGES, RELATIONAL STRUCTURES⟩, there are two obvious variations that should be considered: the relationships with other languages for describing relational structures, and the use of other kinds of structures for interpreting modal languages.

As regards the first option, there are many well-known alternative languages for talking about relational structure: most obviously, first- or second-order classical languages. And indeed, every modal language has *corresponding* classical languages that describe the same class of structures. But although both modal and classical languages talk about relational structures, they do so very differently. Whereas modal languages take an internal perspective, classical languages, with their quantifiers and variable binding, are the prime example of how to take an *external* perspective on relational structures. In spite of this, there is a *standard translation* of any modal language into its corresponding classical language. This translation provides a bridge between the worlds of modal and classical logic, enabling techniques and results to be imported and exported. The resultant study is called *correspondence theory*, and it is a cornerstone of modern modal logic.

In the most important example of the second variation, modal logic is linked up with universal algebra via the apparatus of *duality theory*. In this framework, modal formulas are viewed as algebraic terms which have a natural algebraic semantics in terms of *boolean algebras with operators*, and, from this perspective, modal logic is essentially the study of certain varieties of *equational logic*. Now, even in isolation, this algebraic perspective is of interest – but what makes it a truly formidable tool is the way it interacts with the perspective provided by relational structures. Roughly speaking, relational structures can be constructed out of algebras, and algebras can be constructed out of relational structures, and both constructions preserve essential logical properties. The key technical result that underlies this duality is the Jónsson-Tarski Theorem, a Stone-like representation theorem for boolean algebras with operators. This opens the door to the world of universal algebra and, as we will see, the powerful techniques to be found there lend themselves readily to the analysis of modal logic.

Slogan 3 is fundamental to the way the material in this book is developed: modal logic will be systematically linked to the wider logical world by both correspondence and duality theory. We do not view modal logic as a 'non-classical logic' that studies 'intensional phenomena' via 'possible world semantics.' This is one interpretation of the machinery we will discuss – but the real beauty of the subject lies deeper.

Let us try and summarize our discussion. Modal languages are syntactically simple languages that provide an internal perspective on relational structures. Because of their simplicity, they are becoming increasingly popular in a number of applications. Moreover, modal logic is surprisingly mathematically rich. This richness is due to the intricate interplay between modal languages and the relational structures that interpret them. At its most straightforward, the relational interpretation gives us a natural semantic perspective from which to attack problems directly. But the interplay runs deeper. By adopting the perspective of correspondence theory, modal logic can be regarded as a fragment of first- or second-order classical logic. Moreover, by adopting an algebraic perspective, we obtain a different (and no less classical) perspective: modal logic as equational logic. The fascination of modal logic ultimately stems from the (still not fully understood) links between these perspectives.

What this book is about

This book is a course in modal logic, intended for both novices and more experienced readers, that presents modal logic as a powerful and flexible tool for working with relational structures. It provides a thorough grounding in the basic relational perspective on modal logic, and applies this perspective to issues in completeness, computability, and complexity. In addition, it introduces and develops in some detail the perspectives provided by correspondence theory and algebra.

This much is predictable from our earlier discussion. However, three additional desiderata have helped shape the book. First, we have attempted to emphasize the *flexibility* of modal logic as a tool for working with relational structures. One still encounters with annoying frequency the view that modal logic amounts to rather simple-minded uses of two operators $\Diamond$ and $\Box$. This view has been out of date at least since the late 1960s (say, since Hans Kamp's expressive completeness result for since/until logic, to give a significant, if arbitrary, example), and in view of such developments as propositional dynamic logic and arrow logic it is now hopelessly anachronistic and unhelpful. We strongly advocate a liberal attitude in this book: we switch freely between various modal languages and in the final chapter we introduce a variety of further 'upgrades.' And as far as we are concerned, *it is all just modal logic*.

Second, two pedagogic goals have shaped the writing and selection of material: we want to explicate a range of *proof techniques* which we feel are significant and worth mastering, and, where appropriate, we want to draw attention to some important *general results*. These goals are pursued fairly single mindedly: on occasion, a single result may be proved by a variety of methods, and every chapter (except the following one) proves at least one very general and (we hope) very interesting result. The reader looking for a catalogue of facts about his or her favorite modal system probably will not find it here. But such a reader may well find the technique needed to algebraize it, to analyze its expressive power, to prove a completeness result, or to establish its decidability or undecidability – and may even discover that the relevant results are a special case of something known.

Finally, contemporary modal logic is profoundly influenced by its applications, particularly in theoretical computer science. Indeed, some of the most interesting advances in the subject (for example, the development of propositional dynamic logic, and the investigation of modal logic from a complexity-theoretic standpoint) were largely due to computer scientists, not modal logicians. Such influences must be acknowledged and incorporated, and we attempt to do so.

What this book is not about

Modal logic is a broad field, and inevitably we have had to leave out a lot of interesting material, indeed whole areas of active research. There are two principle omissions: there is no discussion of first-order modal systems or of non-Hilbert-style proof theory and automated reasoning techniques.

The first omission is relatively easy to justify. First-order modal logic is an enterprise quite distinct from the study of propositional systems: its principle concern is how best to graft together classical logic and propositional modal logic. It is an interesting field, and one in which there is much current activity, but its concerns lie outside the scope of this book.

The omission of proof theory and automated reasoning techniques calls for a little more explanation. A considerable proportion of this book is devoted to completeness theory and its algebraic ramifications; however, as is often the case in modal logic, the proof systems discussed are basically Hilbert-style axiomatic systems. There is no discussion of natural deduction, sequent calculi, labeled deductive systems, resolution, or display calculi. A (rather abstract) tableaux system is used once, but only as a tool to prove a complexity result. In short, there is little in this book that a proof theorist would regard as real proof theory, and nothing on implementation. Why is this? Essentially because modal proof theory and automated reasoning are still relatively youthful enterprises; they are exciting and active fields, but as yet there is little consensus about methods and few general results. Moreover, these fields are moving fast; much that is currently regarded as state of the art is likely to go rapidly out of date. For these reasons we have decided – rather reluctantly – not to discuss these topics.

In addition to these major areas, there are a host of more local omissions. One is provability and interpretability logic. While these are fascinating examples of how modal logical ideas can be applied in mathematics, the principle interest of these fields is not modal logic itself (which is simply used as a tool) but the formal study of arithmetic: a typical introduction to these topics (and several excellent ones exist, for example Boolos [67, 68], and Smoryński [416]) is typically about ten percent modal and ninety percent arithmetical. A second omission is a topic that is a traditional favorite of modal logicians: the fine structure of the lattice of normal modal logics in the basic $\Diamond$ and $\Box$ language; we confine ourselves in this book to the relatively easy case of logics extending **S4.3**. The reader interested in learning more about this type of work should consult Bull and Segerberg [74] or Chagrov and Zakharyaschev [87]. Other omissions we regret include: a discussion of meta-logical properties such as interpolation, a detailed examination of local versus global consequence, and an introduction to the modal μ-calculus and model checking. Restrictions of space and time made their inclusion impossible.

Audience and prerequisites

The book is aimed at people who use or study modal logic, and more generally, at people working with relational structures. We hope that the book will be of use to two distinct audiences: a less experienced audience, consisting of students of logic, computer science, artificial intelligence, philosophy, linguistics, and other fields where modal logic and relational structures are of importance, and a more experienced audience consisting of colleagues working in one or more of the above research areas who would like to learn and apply modal logic in their own area. To this end, there are two distinct tracks through this book: the basic track (this consists of selected sections from each chapter, and will be described shortly) and an advanced track (that is, the entire book).

The book starts at the beginning, and does not presuppose prior acquaintance with *modal* logic; but, even on the basic track, prior acquaintance with first-order logic and its semantics is essential. Furthermore, the development is essentially mathematical and assumes that the reader is comfortable with such things as sets, functions, relations and so on, and can follow mathematical argumentation, such as proofs by induction. In addition, although we have tried to make the basic track material as self-contained as possible, two of the later chapters probably require a little more background knowledge than this. In particular, a reader who has never encountered boolean (or some other) algebras before is likely to find Chapter 5 hard going, and the reader who has never encountered the concept of computable and uncomputable problems will find Chapter 6 demanding. That said, only a relatively modest background knowledge in these areas is required to follow the basic track material; certainly the main thrust of the development should be clear. The requisite background material in logic, algebra and computability can be found in Appendices A, B, and C.

Needless to say, we have also tried to make the advanced track material as readable and understandable as possible. However, largely because of the different kinds of background knowledge required in different places, advanced track readers may sometimes need to supplement this book with a background reading in model theory, universal algebra or computational complexity. Again, the required material is sketched in the appendices.

Contents

The chapter-by-chapter breakdown of the material is as follows.

Chapter 1. Basic Concepts. This chapter introduces a number of key modal languages (namely the *basic modal language*, *modal languages of arbitrary similarity type*, the *basic temporal language*, the language of *propositional dynamic logic*, and *arrow languages*), and shows how they are interpreted on various kinds of relational structures (namely *models*, *frames* and *general frames*). It also establishes notation, discusses some basic concepts such as *satisfaction*, *validity*, *logical consequence* and *normal modal logics*, and places them in historical perspective. The entire chapter is essentially introductory; all sections lie on the basic track.

Chapter 2. Models. This chapter examines modal languages as tools for talking about models. In the first five sections we prove some basic *invariance results*, introduce *bisimulations*, discuss the use of *finite models*, and, by describing the *standard translation*, initiate the study of *correspondence theory*. All five sections are fundamental to later developments – indeed the sections on bisimulations and the standard translation are among the most important in the entire book – and together they constitute the basic track selection. The remaining two sections are on the advanced track. They probe the expressive power of modal languages using ultrafilter extensions, ultraproducts, and saturated models; establish the fundamental role of bisimulations in correspondence theory; and introduce the concepts of simulation and safety.

Chapter 3. Frames. This chapter examines modal languages as tools for talking about frames; all sections, save the very last, lie on the basic track. The first three sections develop the basic theory of frame correspondence: we give examples of frame definability, show that relatively simple modal formulas can define frame conditions beyond the reach of any first-order formula (and explain why this happens), and introduce the concepts needed to state the celebrated *Goldblatt-Thomason* Theorem. After a short fourth section which discusses finite frames, we embark on the study of the *Sahlqvist fragment*. This is a large class of formulas, each of which corresponds to a first-order frame condition, and we devote three sections to it. In the final (advanced) section we introduce some further frame constructions and prove the Goldblatt-Thomason Theorem model theoretically.

Chapter 4. Completeness. This chapter has two parts; all sections, save the very last, lie on the basic track. The first part, consisting of the first four sections, is an introduction to basic completeness theory (including *canonical models*, *completeness-via-canonicity* proofs, *canonicity failure*, and *incompleteness*). The second part is a survey of methods that can be used to show completeness when canonicity fails. We discuss *transformation methods*, the *step-by-step* technique, the use of *rules for the undefinable*, and devote the final two sections to a discussion of *finitary methods*. The first of these sections proves the completeness of Propositional Dynamic Logic (**PDL**). The second (the only section on the advanced track) examines extensions of **S4.3**, proving (among other things) Bull's Theorem.

Chapter 5. Algebras and General Frames. The first three sections lie on the basic track: we discuss the role of algebra in logic, show how algebraic ideas can be applied to modal logic via *boolean algebras with operators*, and then prove the fundamental *Jónsson-Tarski Theorem*. With the basics thus laid we turn to *duality theory*, which soon leads us to an algebraic proof of the *Goldblatt-Thomason* Theorem (which was proved model theoretically in Chapter 3). In the two remaining sections (which lie on the advanced track) we discuss *general frames* from an algebraic perspective, introduce the concept of *persistence* (a generalization of the idea of canonicity) and use it to prove the *Sahlqvist Completeness Theorem*, the completeness-theoretic twin of the correspondence result proved in Chapter 3.

Chapter 6. Computability and Complexity. This chapter has two main parts. The first, comprising the first five sections, is an introduction to decidability and undecidability in modal logic. We introduce the basic ideas involved in computing modal satisfiability and validity problems, and then discuss three ways of proving decidability results: the use of *finite models*, the method of *interpretations*, and the use of *quasi-models* and *mosaics*. The fifth section gives two simple examples which illustrate how easily undecidable – and indeed, highly undecidable – modal logics can arise. All of the first part lies on the basic track. The remaining three sections examine modal logic from the perspective of computational complexity. In particular, the modal relevance of three central complexity classes (NP, PSPACE, and EXPTIME) is discussed in some detail. We pay particular attention to PSPACE, proving Ladner's general PSPACE-hardness result in detail. These sections lie on the advanced track, but this is partly because computational complexity is likely to be a new subject for some readers. The material is elegant and interesting, and we have tried to make these sections as self-contained and accessible as possible.

Chapter 7. Extended Modal Logic. This chapter has a quite different flavor from the others: it is essentially the party at the end of the book in which we talk about some of our favorite examples of extended modal systems. We will not offer any advice about what to read here – simply pick and choose and enjoy. The topics covered are: boosting the expressive power of modal languages with the aid of *logical modalities*, completeness-via-completeness proofs in *since/until logic*, naming states with the help of *hybrid logics*, and performing evaluation at sequences of states in *multi-dimensional modal logic*. We also show how to export modal ideas back to first-order logic by defining the *guarded fragment*, and conclude by proving a *Lindström Theorem* for modal logic.

Nearly all sections end with exercises. Each chapter starts with a chapter guide outlining the main themes of the sections that follow. Moreover, each chapter finishes with a summary, and – except the first – with a section entitled Notes. These give references for results discussed in the text. (In general we do not attribute results in the text, though where a name has become firmly attached – for example, Bull's Theorem or Lindenbaum's Lemma – we use it.) The Notes also give pointers to relevant work not covered in the text. The final section of Chapter 1 sketches the history of modal logic, and Appendix D gives a brief guide to textbooks, survey articles, and other material on modal logic.

Teaching the book

The book can be used as the basis for a number of different courses. Here are some suggestions.

Modal Logic and Relational Structures. (1 Semester, 2 hours a week)
All of Chapter 1, all the basic track sections in Chapter 2, and all the basic track sections in Chapter 3. This course introduces modal logic from a semantically oriented perspective. It is not particularly technical (in fact, only Section 2.5 is likely to cause any difficulties), and the student will come away with an appreciation of what modal languages are and the kind of expressivity they offer. It is deliberately one-sided – it is intended as an antidote to traditional introductions.

An Introduction to Modal Logic. (1 Semester, 4 hours a week)
All of Chapter 1, all the basic track material in Chapter 2, the first six or seven sections of Chapter 3, the first six or seven sections of Chapter 4, and the first four sections of Chapter 6. In essence, this course adds to the previous one the contents of a traditional introduction to modal logic (namely completeness-via-canonical models, and decidability-via-filtrations) and includes extra material on decidability which we believe *should* become traditional. This course gives a useful and fairly balanced picture of many aspects of modern modal logic.

Modal Logic for Computer Scientists. (1 Semester, 4 hours a week)
All of Chapter 1, the first four sections of Chapter 2, the first four sections of Chapter 3, the first four sections of Chapter 4 plus Section 4.8 (completeness of **PDL**), all of Chapter 6, and a selection of topics from Chapter 7. In our opinion, this course is more valuable than the previous one, and in spite of its title it is *not* just for computer science students. This course teaches basic notions of modal expressivity (bisimulation, the standard translation, and frame definability), key ideas on completeness (including incompleteness), covers both computability and complexity, and will give the student an impression of the wide variety of options available in modern modal logic. It comes close to our ideal of what a modern, well-rounded introduction to modal logic should look like.

Mathematical Aspects of Modal Logic. (1 Semester, 4 hours a week)
Chapters 1, 2, and 3, the first four sections of Chapter 4, and all of Chapter 5. If you are teaching logicians, this is probably the course to offer. It is a demanding course, and requires background knowledge in both model theory and algebra, but we think that students with this background will like the way the story unfolds.

Modal Logic. (2 Semesters, 4 hours a week)
But of course, there is another option: teach the whole book. Given enough background knowledge and commitment, this *is* do-able in 2 semesters. Though we should confess right away that the course's title is *highly* misleading: once you get to the end of the book, you will discover that far from having learned every-

thing about modal logic, you have merely arrived at the beginning of an unending journey.

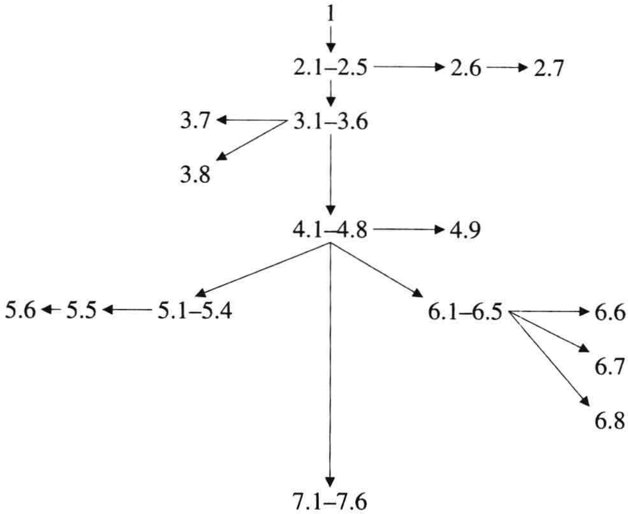

Fig. 1. Dependency Diagram

Hopefully these suggestions will spark further ideas. There is a lot of material here, and by mixing and matching, perhaps combined with judicious use of other sources (see Appendix D, the Guide to the Literature, for some suggestions), the instructor should be able to tailor courses for most needs. The dependency diagram (see Figure 1) will help your planning.

Electronic support

We have set up a home page for this book, where we welcome feedback, and where we will make selected solutions to the exercises and teaching materials available, as well as any corrections that may need to be made. The URL is

`http://www.mlbook.org`

Acknowledgments

We want to thank the following colleagues for their helpful comments and useful suggestions: Carlos Areces, Johan van Benthem, Giacomo Bonanno, Henry Chinaski, Jan van Eijck, Joeri Engelfriet, Paul Gochet, Rob Goldblatt, Val Goranko, Ramon Jansana, Theo Janssen, Tim Klinger, Johan W. Klüwer, Alexander Kurz, Holger Schlingloff, Moshe Vardi, and Rineke Verbrugge. Special thanks are due