4

COMPUTATIONAL MODAL LOGIC

Ian Horrocks, Ullrich Hustadt, Ulrike Sattler, and Renate Schmidt

1 INTRODUCTION

As we have seen in preceding chapters, the worst case complexity of basic reasoning tasks, such as deciding the satisfiability of a modal formula, is at least NP-complete for almost all modal logics. Moreover, for logics extended with features that are useful in practice, the worst case complexity can be much higher, e.g., ExpTime-complete for $\mathbf{K}_n$ extended with non-logical axioms (background theories), and NExpTime-complete for $\mathbf{K}_n$ extended with converse modalities, graded modalities and nominals.

Some may regard these results as discouraging and the question arises whether automated computation with such logics can be feasible in practice. Fortunately, the kinds of pathological formulae/theories that give rise to these worst case results seem to be rarely encountered in realistic applications, and this has allowed for the successful development and deployment of automated reasoning systems for modal logics and their notational variant, description logics; see Chapter 13 of this handbook. Applications of such systems include, e.g., multi-agent systems [53, 60, 196], configuration [137], conceptual modelling [73], information integration [32], and ontology tools and applications [125, 131, 167, 189, 138, 197].

Even for application derived formulae/theories, however, naive implementations of theoretical proof systems, such as the tableau calculi presented in Chapter 2 of this handbook, are unlikely to be of practical utility. As pointed out in [40], without the use of an analytic cut rule, the minimal length of proofs using these calculi can exceed that of proofs using the truth table method for certain propositional (and modal) formulae. Further, not only is it important that short proofs exist, but also how we go about finding a proof or a counter-model. Much of the work presented in this chapter deals with techniques that reduce the size of the search space or help to traverse the search space more efficiently. Successful modern reasoning systems crucially employ specialised reasoning techniques along with optimisations to dramatically improve typical case performance; cf. for example [85, 90, 97, 115, 116, 158, 159]. In this chapter, we focus on reasoning and optimisation techniques used in tableau-based algorithms and translation-based methods.

Translation-based methods make use of the fact that a wide variety of modal logics can be translated into first-order logic; in fact, they can be considered as characterising certain fragments of first-order logic as explained in Section 2 of Chapter 1 of this handbook. To the translated modal formulae, we can apply first-order reasoning methods, in particular, refinements of resolution [16]. Using this combination of a translation method and resolution has some obvious advantages. Any modal logic which can be embedded into first-order logic can be treated. The translations are straightforward, and can be performed in time $O(n \log n)$, so the engineering effort is minimal. For the resolution part, standard resolution provers can be used, or otherwise they can be used with small adaptations. Modern resolution provers [169, 183, 194] are among the most sophisticated and fastest first-order logic theorem provers currently available. The translation method is generic, it can handle first-order modal logics, undecidable modal logics, and combinations of modal and non-modal logics. In all cases, soundness and completeness of the method is immediate from results showing that the translation is satisfiability equivalence preserving and the soundness and completeness of the resolution calculus for first-order logic. The semi-decidability of first-order logic and the behaviour of first-order resolution on first-order formulae does not give us, however, any immediate insight into the modal fragment of first-order logic, which certainly is decidable, or the behaviour of first-order resolution on translated modal formulae. While termination of a resolution derivation from a translated modal formula is not always guaranteed, there are various ways, using different translations and different refinements of resolution, of obtaining translation-based decision procedures. In Section 3, we discuss some of these approaches and illustrate them using the modal logics $\mathbf{K}_n$, $\mathbf{K4}_n$, $\mathbf{KB}_n$, $\mathbf{K}_n^{\smile}$ ($\mathbf{K}_n$ with converse modalities), and $\mathbf{KB4}_n$. Also, using the modal logic $\mathbf{K}_n$, we want to provide some fundamental understanding of how modern resolution provers work in general, what kind of optimisations are available, and how they can be used to provide effective and practical decision procedures for modal logics.

Tableau-based algorithms are closely related to the prefixed tableau systems presented in Section 6 of Chapter 2 of this handbook. In Section 4, we first explain the exact relationship between the two before describing a tableau algorithm which decides the satisfiability of formulae in the basic multi-modal logic $\mathbf{K}_n$. We then discuss implementation and optimisation techniques which can be used to turn this tableau algorithm into an effective and practical decision procedure for $\mathbf{K}_n$. Following the same structure, we also describe tableau-based algorithms for the modal logics $\mathbf{K4}_n$, $\mathbf{K}_n$ with non-logical axioms, $\mathbf{K}_n^{\smile}$, and their combinations and discuss implementation issues of those algorithms. Whereas the $\mathbf{K}_n$ tableau algorithm terminates "automatically", we use certain cycle detection mechanisms to ensure termination for other modal logics. It can be easily seen that these mechanisms must be chosen carefully to preserve correctness of the algorithm and, at the same time, to enable termination as soon as possible so as to avoid an unnecessarily long search. Interestingly, it has been shown by state of the art description logic reasoners [159, 90, 160] that such tableau algorithms are amenable to optimisation, and that they behave better than their worst-case complexity or that of the corresponding reasoning problem suggest: they implement non-deterministic double exponential decision procedures for logics that are ExpTime-complete.

In Section 5, we give an overview of alternative computational approaches to the satisfiability problem in modal logics. These include automata-based algorithms, direct resolution, the inverse method, and sequent-based approaches. In Section 6, we survey reasoning problems other than satisfiability and provability which are relevant for applications of modal logics, namely, model checking, proof checking, and computing correspondence properties for modal axiom schemata. Finally, we conclude the chapter with a brief review and discussion of current and future research.

2 SYNTAX, SEMANTICS, AND REASONING PROBLEMS OF MODAL LOGICS

Throughout this chapter, we use a notation that is compatible with the one presented in Chapter 1 of this handbook. We will use the symbols $p, q, p_i, q_i, \ldots$ for propositional variables. Here, we will be concerned with extensions and variants of the multi-modal logic $\mathbf{K}_n$. The set of $\mathbf{K}_n$ *formulae* is the smallest set that contains all propositional variables, is closed under Boolean operators, and contains $[i]\psi$ and $\langle i\rangle\psi$ for each $1 \leq i \leq n$ and each $\mathbf{K}_n$ formula ψ. Formulae of the form $[i]\psi$ and $\langle i\rangle\psi$ are called *box formulae* and *diamond formulae*, respectively. In different sections, we will consider different normal forms of $\mathbf{K}_n$ formulae, and thus we are generous here and allow all kinds of Boolean operators and abbreviations, e.g. $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\top$ (for any tautology), $\bot$ (for $\neg\top$), etc.

As usual, the semantics of $\mathbf{K}_n$ is defined in terms of relational, Kripke structures or frames. A *frame* is a tuple $\langle W, R\rangle$ of a non-empty set W (of worlds) and a mapping R from natural numbers i, $1 \leq i \leq n$ to binary relations over W, thus $R(i) \subseteq W \times W$. Here and in the rest of the chapter, we use R_i as an abbreviation of $R(i)$, and we say that w is *i-accessible from* v if $R_i(v, w)$. A *model* is given by a triple $\mathfrak{M} = \langle W, R, V\rangle$, where $\langle W, R\rangle$ is a frame and V is a mapping from propositional variables to subsets of W. The notion of a formula ψ *being true* in a model $\mathfrak{M}$ at a world $w \in W$ is inductively defined as follows (we omit the definition for most Boolean operators).