1

MODAL LOGIC: A SEMANTIC PERSPECTIVE

Patrick Blackburn and Johan van Benthem

1 INTRODUCTION

This chapter introduces modal logic from a semantic perspective. That is, it presents modal logic as a tool for talking about *structures* or *models*. But what kind of structures can modal logic talk about?

There is no single answer. For example, modal logic can be given an *algebraic semantics*, and under this interpretation modal logic is a tool for talking about what are known as boolean algebras with operators. And modal logic can be given a *topological semantics*, so it can also be viewed as a tool for talking about topologies. But although we briefly discuss algebraic and topological semantics, for the most part this chapter focuses on modal logic as a tool for talking about *graphs*. To put it another way, this chapter is devoted to what is known as the *relational* or *Kripke* semantics for modal logic. This is the best known and (with the possible exception of algebraic semantics) the best explored style of modal semantics. It is also, arguably, the most intuitive. Over the years modal logic has been applied in many different ways. It has been used as a tool for reasoning about time, beliefs, computational systems, necessity and possibility, and much else besides. These applications, though diverse, have something important in common: the key ideas they employ (flows of time, relations between epistemic alternatives, transitions between computational states, networks of possible worlds) can all be represented as simple graph-like structures. And as we shall see, modal logic is an interesting tool for talking about such structures: it provides an internal perspective on the information they contain.

But modal logic is not the only tool for talking about graphs, and this brings us to one of the major themes of the chapter: the relationship between modal logic and other forms of logic. As we shall see, under the graph-based perspective discussed here, modal logic is closely linked to both first- and second-order classical logic. This immediately raises interesting questions. How does modal logic compare with these logics as a tool for talking about graphs? Can modal expressivity over graphs be characterised in terms of classical logic? We shall ask (and answer) such questions in the course of the chapter.

Games (in various guises) are another recurring motif. The simple way that modal formulas are interpreted on graphs naturally gives rise to games and game-like concepts. The most important of these is the notion of *bisimulation*. This is a relation between two models, weaker than isomorphism, which can be thought of as giving rise to a transition-matching game between two players. As we shall see, this concept holds the key to modal model theory and characterises the link with first-order logic.

This chapter has two pedagogical goals. The first is to provide a bread-and-butter introduction to relational semantics for modal logic that can be used as a basis for tackling the more advanced chapters in this handbook. Thus the reader will find here definitions and discussions of all the basic tools needed in modal model theory (such as the standard translation, generated submodels, bounded morphisms, and so on). Basic results about these concepts are stated and some simple proofs are given. But we have a second, more ambitious, goal: to help the reader start thinking semantically. We want to give the reader a sense of how modal logicians view structure, and what they look for when exploring new logics. To this end we have tried to isolate the intuitions that guide working modal logicians, and to present them vividly. We also make numerous asides, some of which touch on advanced logical topics. Their purpose is to situate the key ideas in a wider context, and even beginners should try to follow them.

Here is our plan. In Section 2, we introduce basic modal languages and the graphs over which they are interpreted. We give the satisfaction definition (which tells us how to interpret modal formulas in graphs) and the standard translation (which links modal logic with classical logic).

With these preliminaries out of the way, we are ready to go deeper. What can (and cannot) modal languages say about graphs? In Section 3 we introduce the notion of bisimulation and use it to develop some answers; among other things, we characterise modal logic as a fragment of first-order logic. In Section 4 we examine the computability and computational complexity of modal logic. A shift of topic? Not at all. In essence, this section examines modal logic as a tool for talking about *finite* graphs. In Section 5 we move to the level of frames and re-examine the link between modal and classical logic. As we shall see, at this level the fundamental correspondence is between modal logic and (monadic) second-order logic. In Section 6 we move beyond the basic modal language and discuss a number of richer languages that offer more expressivity. But what makes them all modal? As we shall see, many of the themes explored in earlier sections re-emerge, and point towards an idea that seems to lie at the heart of modal logic: guarding. Moreover, in some cases it is possible to prove Lindström-style characterisation results. In Section 7 we discuss three alternatives to relational semantics, namely algebraic, neighbourhood, and topological semantics. We conclude in Section 8.

Two final remarks. First, although we introduce modal logic from scratch, we assume that the reader has at least a basic understanding of classical first-order logic (especially its model-theoretic semantics) and some grasp of the notion of computability. Any standard introduction to mathematical logic (Enderton [37] is a good choice) supplies more than enough material to follow the main line of the chapter. Second, we *don't* discuss modal proof-theory or related notions such as completeness in any detail (these topics are the focus of Chapter 2 of this handbook). Although we haven't banished all mention of normal modal logics and completeness from the chapter, in our view traditional introductions to modal logic tend to overemphasise these topics. We want this chapter to act as a counterbalance. As we hope to convince the reader, simply asking the question "But what can I *say* with these languages?" swiftly leads to interesting territory.

2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 First steps in relational semantics

Suppose we have a set of proposition symbols (whose elements we typically write as p, q, r and so on) and a set of modality symbols (whose elements we typically write as m, m', m'', and so on). The choice of PROP and MOD is called the *signature* (or *similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond or a box. There is redundancy in the way we have defined basic modal languages: we don't need all these boolean connectives as primitives, and it will follow from the satisfaction definition given below that, for all $m \in$ MOD, $[m]\varphi$ is equivalent to $\neg\langle m\rangle\neg\varphi$ and $\langle m\rangle\varphi$ is equivalent to $\neg[m]\neg\varphi$ (so boxes and diamonds are what are known as *dual connectives*, just as $\exists$ and $\forall$ are in first-order logic). But we won't bother picking out a preferred set of primitives, as this is not relevant to our discussion. If there is only one modality in our language (that is, if MOD has only one element) we simply write $\Diamond$ and $\Box$ for its diamond and box forms. We often tacitly assume that some signature has been fixed, and say things like "the basic modal language", or "the basic modal language with one diamond". We won't need many syntactic concepts in this chapter, but the following ones will be useful. First, the *subformulas* of a basic modal formula φ are φ itself together with all the formulas used to build φ. Second, we say that a subformula ψ of φ occurs *positively* if it is under the scope of an even number of negations, otherwise we say it occurs *negatively* (when this definition is applied, subformulas of the form $\psi \to \theta$ should be read as $\neg\psi \vee \theta$, and subformulas of the form $\bot$ should be read as $\neg\top$). Finally, the *modal operator depth* of a basic modal formula φ is the maximum level of nesting of modalities in φ, and we write *md*(φ) to denote this number.

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m\in\text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times*, *situations*, *worlds* and other things besides. Each R^m in a model is a binary relation on W, and V is a function (the valuation) that assigns to each proposition symbol p in PROP a subset $V(p)$ of W; think of $V(p)$ as the set of points in $\mathfrak{M}$ where p is true. The first two components $(W, \{R^m\}_{m\in\text{MOD}})$ of $\mathfrak{M}$ are called the *frame* underlying the model. If there is only one relation in the model, we typically write (W, R) for its frame, and (W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m\in\text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

$$
\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m\rangle\varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m]\varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}
$$

A formula φ is *globally satisfied* (*globally true*) in a model $\mathfrak{M}$ if it is satisfied at all points in $\mathfrak{M}$, and if this is the case we write $\mathfrak{M} \models \varphi$. A formula φ is *valid* if it is globally satisfied in all models, and if this is the case we write $\models \varphi$. A formula φ is *satisfiable in a model* $\mathfrak{M}$ if there is some point in $\mathfrak{M}$ at which φ is satisfied, and φ is *satisfiable* if there is some point in some model at which it is satisfied. These definitions are lifted to sets of formulas in the obvious way. For example, a set of basic modal formulas Σ is satisfiable if there is some point in some model at which all the formulas it contains are satisfied. A formula φ is a *semantic consequence* of a set of formulas Σ if for all models $\mathfrak{M}$ and all points w in $\mathfrak{M}$, if $\mathfrak{M}, w \models \Sigma$ then $\mathfrak{M}, w \models \varphi$, and in such a case we write $\Sigma \models \varphi$. Instead of writing $\{\varphi\} \models \psi$ we write $\varphi \models \psi$.

We now have all the concepts needed to begin exploring modal logic. But instead of moving on, let us reflect upon the ideas just introduced. First, note the *internal* character of the modal satisfaction definition: *modal formulas talk about Kripke models from the inside*. In first-order classical logic, when we talk about a model, we do so from the outside. A *sentence* of first-order logic does not depend on the contextual information contained in assignments of values to variables: sentences take a bird's-eye-view of structure, and, irrespective of the variable assignment we use, are simply true or false of a given model. Modal logic works differently: we evaluate formulas *inside* models *at some particular point*. A modal formula is like an automaton placed inside a structure at some point w, and forced to explore by making transitions to accessible points. This may seem a fanciful way of thinking about the satisfaction definition, but it turns out to be crucial. When we isolate the mathematical content of this intuition, we are led, fairly directly, to the notion of *bisimulation*, the key to modal model theory, which we will introduce in Section 3.

Second, note that basic modal languages are syntactically extremely simple: we are working with languages of propositional logic augmented with additional unary operators. And yet these languages clearly pack quantificational punch. Diamonds and boxes can be thought of as macros that encode quantification over R^m-accessible states in a perspicuous variable-free notation. We will shortly define the *standard translation*, which makes this macro analogy precise.

Third, note that Kripke models can (and in our opinion should) be thought of as (directed) graphs. As we have already mentioned, modal logic has been applied in many different areas. What these areas have in common is that they deal with applications in which the important ideas can be represented by relatively simple graph-like structures. Let's consider some examples,

A classic interpretation of Kripke models of the form (W, R, V) is to regard the elements of W as times, and the relation R as the relation of temporal precedence (that is, Rww' means that time w is earlier than time w'). Consider the (directed) graph in Figure 1. This shows a

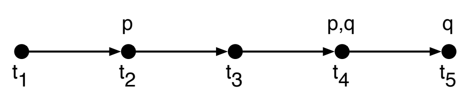

Figure 1. A simple temporal model.

simple flow of time consisting of five points. Here we will take the precedence relation to be the transitive closure of the next-time relation indicated by the arrows (after all, we think of the flow of time as transitive) thus every point t_i precedes all points to its right. Note that (as we would expect from the internal perspective provided by modal languages) whether or not a formula is satisfied depends on where (or in this example, *when*) it is evaluated. For example, the formula $\Diamond(p \wedge q)$ is satisfied at points t_1, t_2 and t_3 (because all these points are to the left of t_4 where both p and q are true together) but not at t_4 and t_5. On the other hand, because q is true at t_5, we have that $\Diamond q$ is true at t_1, t_2, t_3 and t_4. One special case is worth remarking on: note that for any basic formula φ whatsoever, $\Box\varphi$ is satisfied at t_5. Why? Because the clause in the satisfaction definition for boxes says that $\Box\varphi$ is satisfied if and only if φ is satisfied at *all* R-accessible points. As no points are R-accessible from t_5 (it has no points to its right) this condition is trivially met.

The idea of using modal logic as a tool for temporal reasoning is due to Arthur Prior [104, 105]. His work offers what is probably the clearest example of modal logic being appreciated for its internal perspective. In languages such as English and Dutch, the default way of locating information temporally is to use tenses, and tenses locate information *relative* to the point of speech. For example, if at some time t I say "Clarence will fly", then this will be true if at some future time t' Clarence does in fact fly. Prior viewed tensed talk as fundamental: we exist in time, and have to deal with temporal information from the inside. He believed that the internal perspective offered by modal languages made it an ideal tool for capturing the situated nature of our experience and the context-dependent way we talk about it. Prior called his system *tense logic*. He wrote F for the forward looking (or future) diamond, and had a second diamond, written P, for looking back into the past (so in Figure 1, $P(p \wedge q)$ is true at t_5, for this point is to the right of t_4, where p and q are true together). Prior needed backward looking operators to mimic the effect of natural language past tense constructions; for further discussion of Prior's work in this area, see Chapter 19 of this handbook.

Our next example brings us to one of the most influential ways of thinking about Kripke models: to give them a *process interpretation*, which means that we view models as collections of computational states, and the binary relations as computational actions that transform one state into another. This interpretation dates back to the classic work of Hoare [67] and Dijkstra [32]. Let's look at a simple example. Consider the graph shown in Figure 2. This shows a finite state

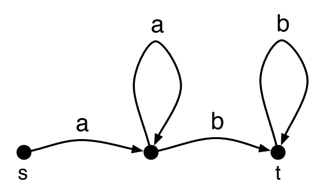

Figure 2. Finite state automaton for $a^n b^m (n, m > 0)$.

automaton for the formal language $a^n b^m$ $(n, m > 0)$, that is, for the set of all strings consisting of a non-empty block of as followed by a non-empty block of bs. But this is precisely the type of graph we can use to interpret a modal language. In this case it would be natural to work with a language with two diamonds $\langle a \rangle$ and $\langle b \rangle$. The $\langle a \rangle$ diamond will be used to explore the a-transitions in the automaton, while the $\langle b \rangle$ diamond explores the b-transitions. It follows that all formulas of the form

$$\langle a \rangle \cdots \langle a \rangle \langle b \rangle \cdots \langle b \rangle t$$

(that is, an unbroken block of $\langle a \rangle$ diamonds preceding an unbroken block of $\langle b \rangle$ diamonds in front of a proposition symbol t which is only true at the terminal node t) are satisfied at the start node s, for all modality sequences of this form correspond to the strings accepted by the automaton. Although simple, this example shows the key feature of many computational interpretations of modal logic: the relations are thought of as processes (here our processes are "read the symbol a" and "read the symbol b"). Note that in this case we are thinking in terms of deterministic processes (each relation is a partial function) but we could just as well work with arbitrary relations, which amounts to working with a non-deterministic model of processes. The process interpretation, in various forms, underlies much of the discussion of this chapter, and it underlies Chapters 12 and 17 of this handbook.

Another important application of modal languages is to model the logic of knowledge and belief; this line of work was pioneered by Jaakko Hintikka [66], and as the more recent treatise by Fagin, Halpern, Moses, and Vardi [39] makes clear, the study of *epistemic logic* continues to flourish. Again, simple graph-based intuitions underly this application. Consider, for example, the graph shown in Figure 3. Here we see the epistemic state of a very simple agent. One

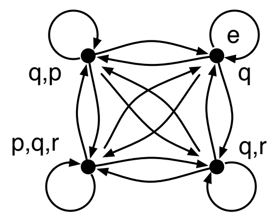

Figure 3. Epistemic state of a simple agent.

of the epistemic situations making up this state is marked e; this represents the agent's current knowledge (the agent knows that q is the case). The other situations represent the way the world might be. For example, although neither p nor r are true in the current situation, the agent views situations in which p and q are true together, and situations in which r and q are true together, and even situations in which p and q and r are all true together, as epistemically acceptable alternatives to the current situation e. So $\Diamond(p \wedge q)$ ("$p \wedge q$ is consistent with what the agent knows"), and $\Diamond(r \wedge q)$, and $\Diamond(p \wedge q \wedge r)$ are all satisfied at e. Moreover $\Box q$ ("the agent knows that q") is satisfied at e, as at every alternative epistemic situation the information q holds. Hintikka introduced the symbol K for this usage of box (that is, he wrote Kq for "the agent knows that q") and his notation is still standard in contemporary epistemic logic. Epistemic logic is discussed in Chapters 18 and 20 of this handbook.

The next example is important for another reason. Modal logic is often viewed as an intrinsically *intensional* logic, interpreted using *possible world semantics*. This view comes from what is probably the most historically influential interpretation of modal logic, namely as the logic of necessity and possibility. In this interpretation, $\Diamond$ is read as "possibly", $\Box$ is read as "necessarily", and the points of Kripke models are regarded as possible worlds. Unfortunately, this interpretation has tended to overshadow the others, at least in certain research communities (some philosophers view modal logic, intensionality, and possible worlds as inextricably intermingled). To ensure that this illusion is dispelled, our last example will be completely *extensional*. Consider the graph in Figure 4.

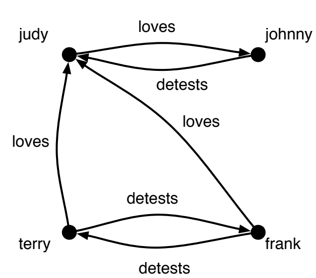

Figure 4. Ordinary individuals.

This is the sort of extensional information that classical logics (such as first-order logic) are often used for. But modal logic is at home here too. We can say lots of interesting things about such situations. For example

$$\langle \text{LOVES} \rangle \top \wedge \langle \text{DETESTS} \rangle \langle \text{LOVES} \rangle \top$$

is true when evaluated at Terry: he loves someone and he detests someone who loves someone. Nowadays, modal logic is widely used for reasoning about such extensional situations. In particular, the concept languages which lie at the heart of the *description logics* used in knowledge representation are often notational variants of (various kinds of) modal languages. Description logics are used in a wide range of applications for representing and reasoning about extensional information. They are treated in depth in Chapter 13 of this handbook.

We're almost ready to define the standard translation, but before doing so let's deal with three other matters. First, in most branches of logic and mathematics, there is a notion of two structures being *isomorphic*, which can be glossed as "mathematically indistinguishable". Let's take this opportunity to be precise about what isomorphism means in basic modal logic (we give the definition for models and frames with one relation; it generalises straightforwardly to structures with multiple relations).

DEFINITION 1 (Isomorphism). Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be models, and $f : W \mapsto W'$ a bijection. If for all $w, v \in W$ we have that Rwv if and only if $Rf(w)f(v)$ then we say that f is an isomorphism between the frames (W, R) and (W', R') and that these frames are isomorphic. If in addition we have, for all proposition symbols p, that $w \in V(p)$ if and only if $f(w) \in V'(p)$ then we say that f is an isomorphism between the models $\mathfrak{M}$ and $\mathfrak{M}'$ and that these models are isomorphic.

As this definition makes clear, if models $\mathfrak{M}$ and $\mathfrak{M}'$ are isomorphic, each replicates perfectly the information in the other. Hence the following result is unsurprising:

PROPOSITION 2. *Let f be an isomorphism between models $\mathfrak{M}$ and $\mathfrak{M}'$. Then for all basic modal formulas φ, and all points w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models \varphi$ if and only if $\mathfrak{M}, f(w) \models \varphi$.*

Proof. Immediate by induction on the construction of φ (see Lemma 9 for an example of such a proof.) ■

Second, we want to point out that it is possible to take a more dynamic perspective on the satisfaction definition. In particular, we can think of it as a game. Let's start with a concrete example. Consider the model in Figure 5.

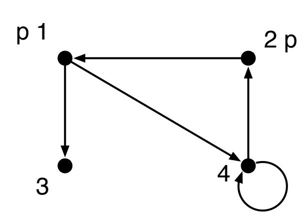

Figure 5. The formula $\Diamond\Box\Diamond p$ is true at 1 and 4, but false at 2 and 3.

As the reader should check, $\Diamond\Box\Diamond p$ is true at points 1 and 4, but false at points 2 and 3. Now suppose we play the following *evaluation game*. This game has two players, a Verifier (V) and a

Falsifier (F), who disagree about the satisfiability of a formula in some model. The two players react differently to the connectives in the formula: for example, occurrences of disjunction allow V to make a choice as to which disjunct to verify, and play continues with the formula chosen; negation switches the roles of the two players; and diamonds make V pick a successor of the current point, while boxes do the same for F. Moreover, for any proposition symbol p, V wins the p-game if p is true at the current state, otherwise F wins. A player also wins the game if the other player must make a move for a modality but cannot.

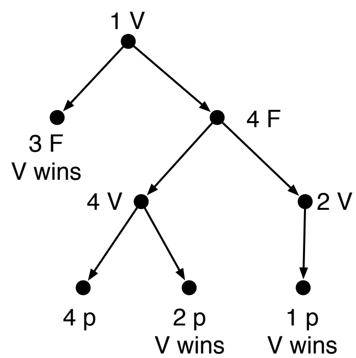

Figure 6. Initial segment of a game tree.

So let's play the game for $\Diamond\Box\Diamond p$ at 1. Figure 6 shows (an initial segment of) the resulting game tree. Note that V can always win. Her most obvious option is to play 3 in response to the outermost diamond; this leaves F with no possible response when faced with the task of falsifying $\Box\Diamond p$. But V can also safely play 4 on her first move. As the tree shows, irrespective of F's response, V can always reach a winning position. What this example suggests is completely general: for any model $\mathfrak{M}$, point w, and basic formula φ, we have that $\mathfrak{M}, w \models \varphi$ if and only if V has a winning strategy when the φ-game is played in $\mathfrak{M}$ starting at w. Moreover, as we see in this example, different strategies correspond to different ways of showing that the given formula is true.

Finally, some historical remarks. Where does the relational interpretation of modal logic come from? The three authors usually cited as pioneers are Saul Kripke, Jaakko Hintikka, and Stig Kanger. Kripke's contributions are the best known (indeed relational semantics is often called Kripke semantics) and Kripke [83, 84] are regarded as landmarks in the development of modal semantics. But Hintikka independently developed the idea in his work on logics of knowledge and belief (see, for example, his classic monograph "Knowledge and Belief" [66]). Furthermore, although his work was not well known at the time, Kanger, in a series of papers and monographs published in 1957, introduced relational semantics for modal logic (see, for example, Kanger [77, 78]). Indeed, the idea of relational semantics seems to have been in the air at around this time, and a number of other logicians (for example Arthur Prior and Richard Montague) discussed similar ideas. For a detailed discussion of who did what and when, the reader should consult Goldblatt [59].

2.2 The standard translation

We now understand what modal languages are, how they can be interpreted in graphs, and why this can be an interesting thing to do. What next? Well, if we were following a traditional path, we would probably remark that as modal languages are to be used for reasoning, some sort of proof system is called for. For example, if we were working in a language with one modality (and in which we had chosen to define $\Diamond$ in terms of $\Box$) we might point out that the set of all modal validities (that is, the *minimal modal logic*) in the language could be axiomatised by a Hilbert-style proof system called **K**. This proof system can be defined in a number ways; we might, for example, stipulate that the axioms of **K** consist of all formulas in the language which have the form of a propositional tautology (by which we mean not merely tautologies such as $p \rightarrow p$ which contain no modalities, but also formulas such as $\Box\Diamond p \rightarrow \Box\Diamond p$, which contain modalities but are truth-functionally tautologous too) and all instances of the following axiom schema:

$$\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi).$$

There are two rules of proof: *modus ponens* (if $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ then $\vdash \psi$) and *modal generalisation* (if $\vdash \varphi$ then $\vdash \Box\varphi$); in the definitions of these rules, $\vdash \theta$ is standard notation that means "the formula θ is provable". Now, this looks like a standard axiomatisation of first-order logic with $\Box$ behaving like $\forall$. But **K** has no analogs of the first-order axioms with tricky side conditions on freedom and bondage of variables, such as $\forall x\varphi \rightarrow [\tau/x]\varphi$, where τ is a first-order term. This is no coincidence. As the standard translation given below will make clear, modal logic is essentially a perspicuous variable-free notation for a fragment of first-order logic.

But proof systems are not our goal. This chapter is concerned with semantic issues, so quite different aspects of modal logic call for our attention. To get the ball rolling, let's return to our basic semantic entities (Kripke models) and ask what they actually are. This will provide a point of entry to one of the main themes of the chapter: the relationship between modal and classical logic.

So, what is a Kripke model? No mystery here. A Kripke model $(W, \{R^m\}_{m\in\text{MOD}}, V)$ is what model theorists call a *relational structure*. That is, we have a domain of quantification W, a collection of binary relations over this domain, and a collection of unary relations as well (after all, $V(p)$ is a unary relation for each $p \in$ PROP). But this means that we are not forced to talk about Kripke models using modal languages: they provide us with everything needed to interpret classical languages too. For example, to talk about a model $(W, \{R^m\}_{m\in\text{MOD}}, V)$ using first-order logic we would simply make use of a first-order language with a binary relation symbol R^m for every $m \in$ MOD, and a unary relation symbol P for every $p \in$ PROP. Modal logicians have a name for this language: they call it the *first-order correspondence language* (for the basic modal language over PROP and MOD).

Why "correspondence language"? Because every basic modal formula (in the language over PROP and MOD) corresponds to a first-order formula from this language via the *standard translation*:

$$\begin{aligned}
\text{ST}_x(p) &= Px\\
\text{ST}_x(\bot) &= \bot\\
\text{ST}_x(\neg\varphi) &= \neg\,\text{ST}_x(\varphi)\\
\text{ST}_x(\varphi\wedge\psi) &= \text{ST}_x(\varphi)\wedge\text{ST}_x(\psi)\\
\text{ST}_x(\langle m\rangle\varphi) &= \exists y(R^m xy \wedge \text{ST}_y(\varphi))\\
\text{ST}_x([m]\varphi) &= \forall y(R^m xy \rightarrow \text{ST}_y(\varphi)).
\end{aligned}$$

That is, the standard translation maps proposition symbols to unary predicates, commutes with booleans, and handles boxes and diamonds by explicit first-order quantification over R^m-accessible points. The variable y used in the clauses for diamonds and boxes is chosen to be any new variable (that is, one that has not been used so far in the translation). We remarked earlier that diamonds and boxes were essentially a simple macro notation encoding quantification over accessible states; the standard translation expands these macros. Note that $\mathrm{ST}_x(\varphi)$ always contains exactly one free variable (namely x). This free variable is what allows the internal perspective, typical of modal logic, to be mirrored in a classical language: assigning a value to this variable is analogous to evaluating a modal formula inside a model at a certain point.

Here's an example of the translation at work:

$$\begin{aligned} \mathrm{ST}_x(p \to \Diamond p) &= \mathrm{ST}_x(p) \to \mathrm{ST}_x(\Diamond p) \\ &= Px \to \mathrm{ST}_x(\Diamond p) \\ &= Px \to \exists y(Rxy \land \mathrm{ST}_y(p)) \\ &= Px \to \exists y(Rxy \land Py). \end{aligned}$$

As the reader can easily check, $p \to \Diamond p$ and its standard translation $Px \to \exists y(Rxy \land Py)$ are equisatisfiable in the following sense: for any model $\mathfrak{M}$, and any point w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models p \to \Diamond p$ if and only if $\mathfrak{M} \models Px \to \exists y(Rxy \land Py)[x \leftarrow w]$, where the notation $[x \leftarrow w]$ means assign w to the free variable x. Unsurprisingly, this relationship is completely general:

PROPOSITION 3. *For any basic modal formula φ, any model $\mathfrak{M}$, and any point w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M} \models \mathrm{ST}_x(\varphi)[x \leftarrow w]$.*

Proof. There is practically nothing to prove. The clauses of the standard translation mirror the clauses of the satisfaction definition. Hence the result is immediate by induction on the structure of modal formulas. ∎

Thus the standard translation gives us a bridge between modal logic and classical logic. And we can immediately use this bridge to transfer meta-theoretic results for first-order logic to modal logic.

PROPOSITION 4. *Basic modal logic has the compactness property. That is, if Σ is a set of basic modal formulas, and every finite subset of Σ is satisfiable, then Σ itself is satisfiable. Moreover, basic model logic has the Löwenheim-Skolem property. That is, if a set of basic modal formulas Σ is satisfiable in at least one infinite model, then it is satisfiable in models of every infinite cardinality.*

Proof. We show that basic modal logic has the Löwenheim-Skolem property. Suppose that Σ is a set of basic modal formulas that has at least one infinite model. Let $\mathrm{ST}_x(\Sigma)$ be the set of (first-order) formulas obtained by standardly translating all the formulas in Σ. Now, as Σ has an infinite model, by Proposition 3 so does $\mathrm{ST}_x(\Sigma)$. But first-order logic has the Löwenheim-Skolem property, hence $\mathrm{ST}_x(\Sigma)$ has a model of every infinite cardinality. But, again by appeal to Proposition 3, each of these models satisfies Σ, so basic modal logic has the Löwenheim-Skolem property too. The argument showing it has the compactness property is similar. ∎

Another easy consequence of the standard translation is that the set of validities (in basic modal languages) is recursively enumerable. For a basic modal formula φ is valid iff $\mathrm{ST}_x(\varphi)$ is a first-order validity, and the set of first-order validities is recursively enumerable.

Let's sum up what we have learned so far. Propositional modal languages are syntactically simple languages that offer a neat (variable-free) notation for talking about relational structures. They talk about relational structures from the inside, using the modal operators to look for information at accessible states. This internal perspective on models, coupled with the simplicity of modal syntax, means that propositional modal logic is an attractive tool for certain applications. Moreover, viewed as a tool for talking about models, any basic modal language can be regarded as a fragment of its corresponding first-order language: the standard translation systematically maps modal formulas to first-order formulas (in one free variable) and makes the quantification over accessible states explicit. This allows us to quickly establish some basic modal meta-theory by appeal to known results for first-order logic.

3 BISIMULATION AND DEFINABILITY

With the basics behind us it is time to look deeper. In particular, it is time to start mapping the expressive strengths and weaknesses of the basic modal language. Now, the expressive power of a language is usually measured in terms of the distinctions it can draw. A language with just the two expressions "like" and "dislike" would provide only the roughest possible classification of the world, whereas a richer language of assent and dissent would make it possible to draw finer distinctions inside the accepted and rejected situations. So what distinctions can modal languages draw? In this section we discuss this question at the level of models, and in Section 5 we shall reconsider it at the level of frames. In what follows it will often be useful to think in terms of *pointed models*. That is, we shall often present models together with an explicit distinguished point to indicate where we are trying to find a difference.

3.1 Drawing distinctions

A modal language (and indeed any logical language whose formulas form a set) can distinguish between some models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$, but not between all such pairs. For example, our basic modal language can distinguish the pair of models shown in Figure 7.

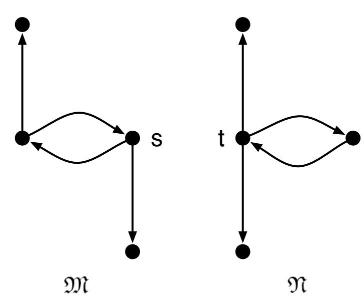

Figure 7. $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ are modally distinguishable.

Here $\Box(\Box\bot \vee \Diamond\Box\bot)$ is a modal formula that distinguishes these models: it is true in $\mathfrak{M}$ at s, but false in $\mathfrak{N}$ at t. But now consider the pair of models shown in Figure 8. Is it possible to *modally* distinguish $(\mathfrak{M}, s)$ from $(\mathfrak{K}, u)$? That is, is it possible to find a (basic) modal formula that is true in $\mathfrak{M}$ at s, but false in $\mathfrak{K}$ at u? Note that it is easy to distinguish them if we are

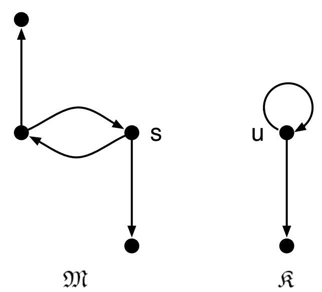

Figure 8. $(\mathfrak{M}, s)$ and $(\mathfrak{K}, u)$ are not modally distinguishable.

allowed to use first-order logic: all points in $\mathfrak{M}$ (including s) are irreflexive, while point u in $\mathfrak{K}$ is reflexive, hence the first-order formula Rxx is not satisfiable (under any variable assignment) in model $\mathfrak{M}$, but it is satisfied in $\mathfrak{K}$ when u is assigned to x. But no matter how ingenious you are, you will not find any formula in the basic modal language that distinguishes these models at their designated points. Why is this?

3.2 Bisimulation

A natural approach to this question is to consider its dual: when should two models be viewed as modally identical? For example, given a process interpretation, when would we view two transition diagrams as representations of the same process? The models $\mathfrak{M}$ and $\mathfrak{K}$ of Figure 8 provide an intuitive example: they seem to stand for the same process when we look at possible actions and deadlocks (note that at each state the process can enter a deadlock situation; that is, it can enter a state from which it cannot exit). By contrast, $\mathfrak{M}$ and $\mathfrak{N}$ in Figure 7 are different, as the right hand state in $\mathfrak{N}$ is not threatened with immediate dead-lock. Or consider the epistemic interpretation: when would we want to say that two graphs represent the same epistemic state? For example, we would probably want to identify the two epistemic models shown in Figure 9 at their distinguished points s and t.

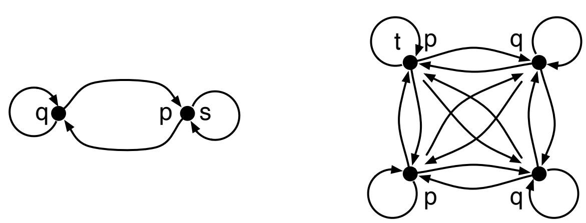

Figure 9. Two epistemically equivalent models.

After all, in essence both models present us with a two way choice: either we are in an epistemic situation where p holds and there is an accessible epistemic situation where q holds, or we are in an epistemic situation where q holds and there is an accessible epistemic situation where p holds. The intuition that both these graphs code the same epistemic state is captured by our modal language: the reader will not find any modal formula that distinguishes them.

The modal logician's idea of asking when two distinct structures are modally identical (that is, make the same modal formulas true) lies within an older (and broader) tradition of looking for the structure preserving morphisms in a given mathematical domain, and letting the corresponding theory describe those notions that are invariant for such morphisms. This is the spirit of Klein's Program in geometry, proposed around 1870, and still influential in many fields. Of course, there is no unique answer to the question of when two structures are the same. This insight was stated forcefully in recent years by President Clinton during the Lewinsky hearings: *It all depends on what you mean by "is"*. Clinton's Principle for modal logic means that we should first try to stipulate some notion of structural equivalence for models that is appropriate for modal languages. This is the purpose of the following definition (first formulated in van Benthem [128, 131]). We state it here for models with one relation R, but the definition generalises straightforwardly to models with any number of relations.

DEFINITION 5 (Bisimulation). A bisimulation between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols,

Zig: if Rwv, then there exists a point v' (in $\mathfrak{M}'$) such that vEv' and $R'w'v'$, and

Zag: if $R'w'v'$, then there exists a point v (in $\mathfrak{M}$) such that vEv' and Rwv.

If there is a bisimulation between two models $\mathfrak{M}$ and $\mathfrak{N}$, then we say that $\mathfrak{M}$ and $\mathfrak{N}$ are bisimilar. Moreover, we say that two states are bisimilar if they are related by some bisimulation.

Putting this in words: two states are bisimilar if they make the same atomic information true and if, in addition, their transition possibilities match. That is, if a transition to a related state is possible in one model, then the bisimulation must deliver a matching transition possibility in the other. Atomic harmony, coupled with the matching transitions concept embodied in the zigzag clauses, make bisimulation a natural notion of process equivalence, and indeed bisimulations were independently discovered in computer science (see Park [100]).

Returning to the models $\mathfrak{M}$, $\mathfrak{K}$, and $\mathfrak{N}$ considered above (and disregarding proposition symbols) it is easy to see that $\mathfrak{M}$ and $\mathfrak{K}$ are bisimilar: the dotted lines in Figure 10 indicate the required bisimulation (note that the indicated bisimulation links the two designated points). Furthermore, it is easy to see that there is no bisimulation that links the designated points of $\mathfrak{N}$ and $\mathfrak{K}$. Why not? Because a move from t to the right-hand world in $\mathfrak{N}$ has no matching move in $\mathfrak{K}$: moving downwards from u is no option (end-points never bisimulate with points having successors) but neither is moving reflexively from u to itself (as one can move from u to a successor which is an endpoint, but this can't be done from the right-hand world in $\mathfrak{N}$).

Given any modal model $\mathfrak{M}$, bisimulations can be used in a number of ways. The so-called *bisimulation contraction* makes $\mathfrak{M}$ as small as possible. To define this, note that it follows from Definition 5 that any union of bisimulations between two models is itself a bisimulation. Therefore the union of all bisimulations between two models is a maximal bisimulation between them. Now form the maximal bisimulation of model $\mathfrak{M}$ with itself (incidentally, a bisimulation of a model with itself is called an *autobisimulation*). Define a quotient of $\mathfrak{M}$ whose points are the equivalence classes, and relate the equivalence class $|w|$ to the equivalence class $|v|$ iff $|w|$ and $|v|$ contain points w' and v' such that $Rw'v'$. The map from points to their equivalence classes is a bisimulation. For example, the bisimulation shown in Figure 10 between $\mathfrak{M}$ and $\mathfrak{K}$ is a bisimulation contraction. Bisimulation contractions are the most compact representation of processes,

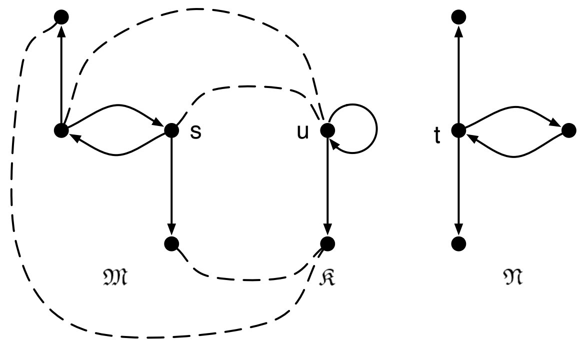

Figure 10. $(\mathfrak{M}, s)$ and $(\mathfrak{K}, u)$ are bisimilar, $(\mathfrak{K}, u)$ and $(\mathfrak{N}, t)$ are not.

at least from a modal standpoint. They remove all the redundancies in the representation — but also all aesthetic symmetries. (A butterfly is a redundant object, as one wing contains enough information under this perspective.)

Bisimulations can also be used to make bigger models: one important construction which does this is called *tree unraveling* (for a very early paper using this construction, see Dummett and Lemmon [34]; for an influential paper that made heavy use of it, see Sahlqvist [111]).

To unravel a model, take all finite R-sequences of points in $\mathfrak{M}$ that start at some point w. These sequences form a tree with one-step extensions of sequences as the tree-successor relation. Projection from a sequence to its last element is a bisimulation onto the original model $\mathfrak{M}$. As an example, consider the unraveling of the two element model $\mathfrak{K}$ around its distinguished point u to the infinite comb-like structure shown in Figure 11 (we use v as the name of the other point in this model). Reasoning about trees is often easier than reasoning about arbitrary graphs, and

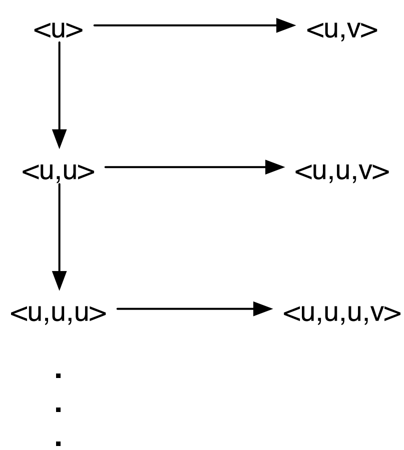

Figure 11. Unraveling $\mathfrak{K}$ around u.

so this method is of considerable theoretical utility. Moreover, as we shall see in the following section, tree unraveling is relevant to the *decidability* of modal logic.

Three other model constructions used in modal logic, namely *disjoint unions*, *generated submodels*, and *bounded morphisms* (or *p-morphisms*) are also bisimulations. Historically, all three constructions were widely used in modal logic more than a decade before the unifying concept of bisimulation was introduced (the classic source for these constructions is Segerberg [113], where they are heavily used, often in combination, to prove completeness theorems). All three constructions are fundamental tools in many areas of modal logic (for example, when reformulated at the level of frames, they are key ingredients in the Goldblatt-Thomason Theorem which we discuss in Section 5) so we take this opportunity to define them for models with one accessibility relation. These definitions generalise straightforwardly to models of arbitrary signature.

The simplest construction is forming disjoint unions. If we have a pair of disjoint models (that is, a pair of models (W, R, V) and (W', R', V') such that W and W' are disjoint) then their disjoint union is the model $(W \cup W', R \cup R', V + V')$, where $V + V'$ is the valuation defined by $(V + V')(p) = V(p) \cup V'(p)$, for all proposition symbols p. That is, forming a disjoint union of two models means lumping together all the information in the two graphs. What if the graphs are not disjoint? Then we simply take disjoint isomorphic copies of the two models, and form the disjoint union of the copies. This lumping together process can be generalised to arbitrarily many models, which prompts the following definition.

DEFINITION 6 (Disjoint Unions). Given mutually disjoint models $\mathfrak{M}_i = (W_i, R_i, V_i)$, where i ranges over the elements of some index set I, we define the disjoint union of these models to be $\mathfrak{M} = (W, R, V)$, where $W = \bigcup_{i \in I} W_i$, $R = \bigcup_{i \in I} R_i$, and $V(p) = \bigcup_{i \in I} V_i(p)$ for all proposition symbols p. To form the disjoint union of a collection of models that are not mutually disjoint, we first take mutually disjoint isomorphic copies, and then form the disjoint union of the copies.

It is immediate from this definition that any component model $\mathfrak{M}_i$ of a disjoint union $\mathfrak{M}$ is bisimilar with $\mathfrak{M}$: for the bisimulation relation E we simply take the identity relation. Identity clearly satisfies the atomic harmony and zigzag conditions required of bisimulations.

Disjoint unions build bigger models from (collections of) smaller ones. Generated submodels do the reverse. They arise by restricting attention to subgraphs of a given graph that are closed under relational transitions. For example, consider the two graphs in Figure 12. It is clear that

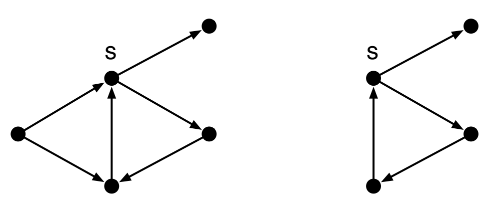

Figure 12. Generating a submodel from s.

the graph on the right arises by restricting attention to a certain transition-closed subgraph of the graph on the left, namely the set of point reachable by taking sequences of transitions from s. This motivates the following definition.

DEFINITION 7 (Generated Submodels). Let $\mathfrak{M} = (W, R, V)$ be a model and let $W' \subseteq W$. We say that a model $\mathfrak{M}' = (W', R', V')$ is the restriction of $\mathfrak{M}$ to W' if $R' = R \cap (W' \times W')$ and for all proposition symbols p we have that $V'(p) = V(p) \cap W'$. We say that W' is R-closed if for all $u \in W'$, if Ruv then $v \in W'$. Finally, we say that $\mathfrak{M}'$ is a generated submodel of $\mathfrak{M}$ iff $\mathfrak{M}'$ is the restriction of $\mathfrak{M}$ to an R-closed subset of W.

If $\mathfrak{M}' = (W', R', V')$ is a generated submodel of $\mathfrak{M} = (W, R, V)$, and $S \subseteq W'$ has the property that every $w' \in W'$ is reachable via a finite sequence of R-transitions from some $s \in S$, then we say that $\mathfrak{M}'$ is the submodel of $\mathfrak{M}$ generated by S. If S is a singleton set $\{s\}$, then we say that $\mathfrak{M}'$ is the submodel of $\mathfrak{M}$ generated by the point s.

A generated submodel is bisimilar to the model that gave rise to it: as with disjoint unions, the identity relation relates the two models in the appropriate way. Incidentally, note that every component model of a disjoint union is a generated submodel of the disjoint union.

Finally we turn to bounded morphisms (or p-morphisms as they are often called).

DEFINITION 8 (Bounded Morphisms). A bounded morphism between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a function f with domain W and range W' such that:

Atomic harmony: Points in W and their f-images satisfy the same proposition symbols (that is, $w \in V(p)$ iff $f(w) \in V'(p)$, for all proposition symbols p).

Morphism: if Rwv, then $R'f(w)f(v)$.

Zag: if $R'f(w)v'$, then there exists a v (in $\mathfrak{M}$) such that $f(v) = v'$ and Rwv.

If f is a bounded morphism from $\mathfrak{M}$ to $\mathfrak{M}'$ and f is surjective, then we say that $\mathfrak{M}'$ is a bounded morphic image of $\mathfrak{M}$.

Bounded morphisms are bisimulations: a bounded morphism is simply a bisimulation in which the bisimulation relation E is an R-preserving morphism f (note that the only essential difference between the two definitions is that the morphism clause replaces the zig clause, and clearly morphism implies zig). Historically, it was the definition of bounded morphisms that inspired the definition of bisimulations.

As an example of a bounded morphism between models, consider Figure 13 (again we ignore proposition symbols).

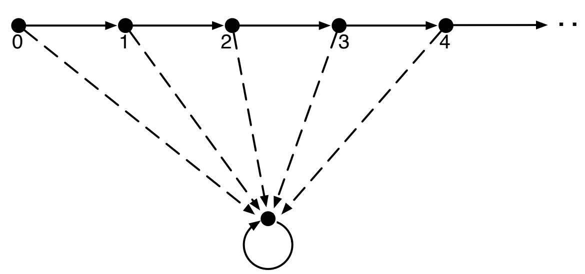

Figure 13. Bounded morphism collapsing the natural numbers to a reflexive point.

Here we have collapsed the natural numbers in their usual order to a single reflexive point. It is clear that this map satisfies both the morphism and zig clauses, so it is indeed a bounded morphism.

3.3 Invariance and definability in first-order logic

Structural invariances preserve certain patterns definable in appropriate languages. Before pursuing the match between bisimulation and modal logic, let us examine the situation in first-order logic. The archetypal structural invariance is *isomorphism* between models. As we saw earlier (recall Proposition 2) modal formulas are invariant for isomorphism. More generally, it is well known that if f is an isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$, then for each first-order formula $\varphi(x_1, \ldots, x_k)$, and each matching tuple of objects $\langle d_1, \ldots, d_k \rangle$ in $\mathfrak{M}$, the following equivalence holds:

$$\mathfrak{M} \models \varphi[d_1, \ldots, d_k] \text{ \textit{iff} } \mathfrak{N} \models \varphi[f(d_1), \ldots, f(d_k)],$$

or stated in words: first-order formulas are invariant for isomorphism.

On special models, the converse also holds. For example, it is a well-known fact that any two finite models with the same first-order theory are isomorphic. But no general converse holds, as there are many more isomorphism classes of models than complete first-order theories. Invariance for isomorphism is even a defining condition for any logic in abstract model theory. But no matter how strong the logic, the converse still fails whenever the formulas of a logic form a set, as opposed to the proper class of isomorphism types.

Thus it makes sense to look at invariance conditions for weaker notions of structural equivalence. For example, a *potential isomorphism* between two models $\mathfrak{M}$ and $\mathfrak{N}$ is a non-empty set I of finite partial isomorphisms satisfying the back-and-forth extension conditions that, whenever $f \in I$ and $d \in \mathfrak{M}$, then there is an $e \in \mathfrak{N}$ such that $f \cup \{(d, e)\} \in I$, and vice-versa. Note that isomorphisms induce potential isomorphisms: simply take I to be the family of all finite restrictions. The converse is not true. Matching up all finite sequences of rational numbers with equally long sequences of real numbers (in the same order) is a potential isomorphism between $\mathbb{Q}$ and $\mathbb{R}$, even though these two structures are not order-isomorphic for cardinality reasons.

It is easy to show that all first-order formulas are invariant for potential isomorphism, but the real match is with a stronger language: two models are potentially isomorphic iff they have the same complete theory in the *infinitary* first-order logic $\mathcal{L}_{\infty\omega}$. This formalism also gives rise to much stronger definability results. For example, for each model $\mathfrak{M}$ there is a sentence $\delta_{\mathfrak{M}}$ of $\mathcal{L}_{\infty\omega}$ which holds only in those models $\mathfrak{N}$ which have a potential isomorphism with $\mathfrak{M}$; that is, models can be defined up to potential isomorphism. Moreover, countable models can even be defined (modulo isomorphism) using only countable conjunctions and disjunctions. This is all very nice of course, but infinitary logic is a bit outlandish from a practical viewpoint.

Better matches between structural invariance and first-order definability arise in the more fine-grained setting of Ehrenfeucht-Fraïssé comparison games between models $\mathfrak{M}$ and $\mathfrak{N}$ played between a Spoiler (who looks for differences between the models) and a Duplicator (who looks for analogies between them). Models $\mathfrak{M}$ and $\mathfrak{N}$ have the same first-order theory up to quantifier depth k iff the Duplicator has a winning strategy in their comparison game over k rounds. We won't give details here, as we will define a modal comparison game of this sort at the end of the section.

3.4 Invariance and definability in modal logic

With these analogies in mind, let us now investigate the modal situation. For a start, modal formulas are *invariant for bisimulation*:

LEMMA 9 (Bisimulation Invariance Lemma). *If E is a bisimulation between $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$, and wEw', then w and w' satisfy the same basic modal formulas.*

Proof. By induction on the construction of modal formulas. The case for proposition symbols is immediate by atomic harmony. The inductive steps for the boolean connectives are straightforward. And the inductive step for $\Diamond$ formulas shows exactly what the zigzag clauses were designed for. For consider the left to right direction. Given $\mathfrak{M}, w \models \Diamond\varphi$ and wEw', we want to show that $\mathfrak{M}', w' \models \Diamond\varphi$. Now, $\mathfrak{M}, w \models \Diamond\varphi$ means that there is some v in $\mathfrak{M}$ such that Rwv and $\mathfrak{M}, v \models \varphi$. But then (by zig) there must be a point v' in $\mathfrak{N}'$ such that vEv' and $R'w'v'$. By the induction hypothesis, $\mathfrak{M}', v' \models \varphi$, hence $\mathfrak{M}', w' \models \Diamond\varphi$ as required. The argument for the right to left direction is essentially the same, using zag in place of zig. ■

The result allows us to show failures of bisimulation easily. For example, we have already sketched an argument showing that the models $\mathfrak{N}$ and $\mathfrak{K}$ of Figure 10 have no bisimulation between their designated points, but a quicker proof is now possible: these points *cannot* be bisimilar because there are modal formulas (for example $\Box(\Box\bot \vee \Diamond\Box\bot)$) which are satisfied at one point but not the other. On the other hand, the dotted lines in Figure 10 show that $\mathfrak{M}$ and $\mathfrak{K}$ are bisimilar; it follows that all points linked by a dotted line in these graphs make exactly the same modal formulas true. Another typical application of this result is to show the undefinability of certain structural notions. For example, we can show that irreflexivity is modally undefinable: no modal formula holds in exactly those points w of models such that $\neg Rww$. To prove this, it suffices to find two bisimilar points in two models, one of which is reflexive, the other irreflexive. One such example is the bisimulation between the designated points of $\mathfrak{M}$ and $\mathfrak{K}$ shown in Figure 10. Another is the bounded morphism of Figure 13 which collapses the natural numbers to a single reflexive point.

Another consequence of this result is that the disjoint union, generated submodel, and bounded morphism constructions are all satisfaction preserving. More precisely:

LEMMA 10. *Modal satisfaction is invariant under the formation of disjoint unions, generated submodels, and bounded morphisms. That is:*

1. *If $\mathfrak{M} = (W, R, V)$ is the disjoint union of $\mathfrak{M}_i = (W_i, R_i, V_i)$, for i from some index set I, then for all $w \in W_i$ and all $i \in I$ we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}_i, w \models \varphi$.*
2. *If $\mathfrak{M}' = (W', R', V')$ is a generated submodel of $\mathfrak{M} = (W, R, V)$, then for all $w' \in W'$ we have that $\mathfrak{M}, w' \models \varphi$ iff $\mathfrak{M}', w' \models \varphi$.*
3. *If $\mathfrak{M}' = (W', R', V')$ is a bounded morphic image of $\mathfrak{M} = (W, R, V)$ under the bounded morphism f, then for all $w \in W$ we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}', f(w) \models \varphi$.*

Proof. All three results could be proved by induction on the structure on φ. But such proofs are unnecessary: we know that disjoint unions, generated submodels, and bounded morphisms are all examples of bisimulations, hence these results follow from Lemma 9. ■

To sum up the discussion so far, bisimulation implies modal equivalence. But what about the converse? For finite models, we have the following.

PROPOSITION 11. *If points w and w' from two finite models $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same modal formulas, then there is a bisimulation E between $\mathfrak{M}$ and $\mathfrak{N}$ such that wEw'.*

Proof. Assume we are working with models containing only a single relation R. We will show that the relation of modal equivalence is itself a bisimulation. That is, we will define the bisimulation relation E by wEw' iff w and w' make the same modal formulas true. We now verify that E so defined is indeed a bisimulation.

It is immediate that E satisfies atomic harmony. As for zig, assume that wEw' and Rwv. Assume for the sake of contradiction that there is no v' in $\mathfrak{M}'$ such that $R'w'v'$ and vEv'. Let $S' = \{u' \mid R'w'u'\}$. Now, as w has an R-successor v, we have $\mathfrak{M}, w \models \Diamond\top$. As wEw', we have $\mathfrak{M}', w' \models \Diamond\top$ too, hence S' is non-empty. Furthermore, as $\mathfrak{M}'$ is finite, S' must be finite too, so we can write it as $\{u'_1, \ldots, u'_n\}$. By assumption, for every $u'_i \in S'$ there exists a formula ψ_i such that $\mathfrak{M}, v \models \psi_i$ but $\mathfrak{M}', u'_i \not\models \psi_i$. It follows that

$$\mathfrak{M}, w \models \Diamond(\psi_1 \wedge \cdots \wedge \psi_n) \text{ and } \mathfrak{M}', w' \not\models \Diamond(\psi_1 \wedge \cdots \wedge \psi_n),$$

which contradicts our assumption that wEw'. Hence E satisfies zig. A symmetric argument shows that E satisfies zag too, hence it is a bisimulation. ■

Thus, on finite models, the expressive power of modal languages matches up exactly with bisimulation invariance. This result can be extended to broader model classes, such as models with finite branching width for successors (note that the proof just given does not depend on the models involved being finite: it would also work for infinite models in which each point has only finitely many R-successors) and suitably saturated models in a model-theoretic sense. But no general converse can hold, for the set-theoretic reasons mentioned earlier. Indeed, the converse does not hold generally even for countable models: not all modally equivalent countable models are bisimilar. Consider the two models in Figure 14 (assume that all proposition symbols are true at all points in both models). Both models have infinitely many branches leading away from their root nodes, but whereas all the branches in the model on the left are of finite length, the model on the right has a branch of infinite length. Now, as the reader should check, both models satisfy the same modal formulas at their root nodes. However there is no bisimulation that links their root nodes; the infinite branch in the model on the right makes it impossible to define one.

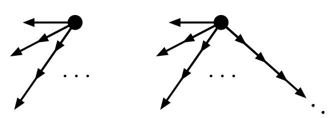

Figure 14. Modally equivalent but not bisimilar.

This counterexample could be repaired by passing to an *infinitary modal language* $\mathcal{L}^{\Diamond}_{\infty\omega}$ with arbitrary (countable) conjunctions and disjunctions. Infinitary modal equivalence occurs between countable models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ whenever there is a bisimulation linking s to t. Furthermore, every countable model $(\mathfrak{M}, s)$ is defined up to bisimulation by some $\mathcal{L}^{\Diamond}_{\infty\omega}$ formula $\delta_{\mathfrak{M},s}$. Again, such infinitary languages are somewhat impractical, but there are some useful bisimulation invariant formalisms which lie between the basic modal language and its infinitary extension. Two examples are *propositional dynamic logic* and the *modal μ-calculus*, which are discussed in Section 6.

Lemma 9 and its partial converses do not exhaust what needs to be said about the role played by bisimulations in modal model theory. But to gain a deeper understanding, we need to bring in a third component: the first-order correspondence language that we met in Section 2.2 when we introduced the standard translation.

3.5 Modal logic and first-order logic compared

The basic modal language can be viewed as a sort of miniature version of full first-order logic over graph models. The standard translation defined in the previous section shows that each modal formula φ corresponds to a first-order formulas $\text{ST}_x(\varphi)$ containing a free variable x. But the converse does not hold: some first-order formulas in the correspondence language are not modally definable. We have already see an example. As the bisimulation between models $\mathfrak{M}$ and $\mathfrak{K}$ shows (recall Figure 10) no modal formula defines $\neg Rxx$. Thus, viewed as a tool for talking about models, modal logic is strictly less expressive than the full first-order correspondence language. And this prompts a further question: given that a modal language is essentially a fragment of the corresponding first-order language, exactly which fragment is it? This question has an elegant answer. First, a preliminary definition.

DEFINITION 12. A first-order formula $\varphi(x)$ is invariant for bisimulation if for all models $\mathfrak{M}$ and $\mathfrak{M}'$, and all points w in $\mathfrak{M}$ and w' in $\mathfrak{M}'$, and all bisimulations E between $\mathfrak{M}$ and $\mathfrak{M}'$ such that wEw', we have that $\mathfrak{M} \models \varphi[x \leftarrow w]$ iff $\mathfrak{M}' \models \varphi[x \leftarrow w']$.

We can now state the main result: basic modal languages correspond to the fragment of their first-order correspondence language that is invariant for bisimulation. More precisely:

THEOREM 13 (Modal Characterisation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$ in one free variable x:*

1. *$\varphi(x)$ is invariant for bisimulation.*
2. *$\varphi(x)$ is equivalent to the standard translation of a basic modal formula.*

Proof. That clause 2 implies 1 is a more or less immediate consequence of Lemma 9. The hard direction is showing that clause 1 implies 2. The original proof can be found in van Benthem [128, 131]. Two other proofs are given in Chapter 5 of this handbook. One is quite close to van Benthem's original approach, the other is based on games. ■

Nowadays many different proofs are known for this result, and for various extensions and variants. For example, Rosen [109] showed that the result holds over finite models; this is far from obvious, as the restriction to finite models means that many standard results of first-order model theory (such as the Compactness Theorem) cannot be applied. And Otto [99] showed that the modal equivalent guaranteed to exist by the previous theorem can be restricted to a formula of modal operator depth 2^k, where k is the quantifier depth of $\varphi(x)$.

Basic modal logic and first-order logic are analogous in many ways. As we mentioned in Section 2, via the standard translation modal logic immediately inherits basic meta-theoretic properties of its more powerful neighbour, such as the Compactness and Löwenheim-Skolem Theorems. But not all such transfer is automatic. Consider, for example, the *Craig Interpolation* property:

> *If $\varphi \models \psi$ then there exists a formula θ whose vocabulary is included in that of both φ and ψ such that $\varphi \models \theta$ and $\theta \models \psi$.*

Does the same result hold for basic modal formulas φ and ψ such that $\varphi \models \psi$? Appealing to the result for first-order logic gives us a first-order formula θ such that $\text{ST}_x(\varphi) \models \theta$ and $\theta \models \text{ST}_x(\psi)$. But what guarantees that this interpolant is modally definable? Interpolation does in fact hold for the basic modal language (for a detailed account, see Chapter 8 of this handbook), but additional work is needed to prove this. Nonetheless, interpolation does mesh well with the above preservation results; here is an improvement on the Modal Characterisation Theorem. We say that a first-order formula φ *implies* ψ *along bisimulation* if the following implication holds: if E is a bisimulation between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$, and $\mathfrak{M}, s \models \varphi$, then $\mathfrak{N}, t \models \psi$.

THEOREM 14 (Modal Characterisation-Interpolation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$:*

1. $\varphi(x)$ *implies* $\psi(x)$ *along bisimulation.*

2. *There is a modally definable θ in the common vocabulary of φ and ψ such that $\varphi \models \theta$ and $\theta \models \psi$.*

Proof. The proof can be found in Barwise and van Benthem [11]. Note that the Modal Characterisation Theorem follows by taking $\varphi(x)$ equal to $\psi(x)$. This result does not imply ordinary modal interpolation as it stands: additional work is again needed. ■

Behind the above observations is the fact that the cheaply transferred properties are universal in some sense, whereas the universal-existential property of interpolation requires honest work. Even so, there is an intuition (based on decades of positive experience with transferring results) that modal logic and first-order logic share all general meta-properties except decidability. No proofs of significant formulations of this idea have been found so far, but we can point to some broad analogies regarding methods. Generally speaking, bisimulation plays the same role for modal logic that potential isomorphism does for first-order logic. This can even be made precise in the following sense. To each first-order model $\mathfrak{M}$ we can associate a modal model whose points are the variable assignments into $\mathfrak{M}$, and whose accessibility relations are changes from one assignment g to another $g(x := d)$ that resets the value for the variable x to the object $d \in \mathfrak{M}$. Then two models $\mathfrak{M}$ and $\mathfrak{N}$ have a potential isomorphism between them iff their associated modal models are bisimilar; see van Benthem [136] for details.

We conclude this discussion with two general results that allow us to switch between modal and first-order relations between models. In essence, both results have the form of a commutative diagram.

LEMMA 15 (First Lifting Lemma). *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

1. $(\mathfrak{M}, s)$ *and* $(\mathfrak{N}, t)$ *are modally equivalent.*

2. $(\mathfrak{M}, s)$ *and* $(\mathfrak{N}, t)$ *have elementary extensions to models* $(\mathfrak{M}^+, s)$ *and* $(\mathfrak{N}^+, t)$ *which are bisimilar.*

LEMMA 16 (Second Lifting Lemma). *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

1. $(\mathfrak{M}, s)$ *and* $(\mathfrak{N}, t)$ *are modally equivalent.*

2. $(\mathfrak{M}, s)$ *and* $(\mathfrak{N}, t)$ *are bisimilar to models* $(\mathfrak{M}^+, s)$ *and* $(\mathfrak{N}^+, t)$ *which are elementarily equivalent.*

Proof. The first lifting lemma was originally proved in van Benthem [128]. It is the key item in (some proofs of) the Characterisation Theorem (the $^+$-models are suitably saturated elementary extensions which allow the Characterisation Theorem to be proved rather straightforwardly). The second lifting lemma (see van Benthem [134] for the original result, and Andréka, van Benthem, and Németi [5] for full proof details) involves judicious tree unraveling of the two models, duplicating sub-trees to create uniformity, coupled with an Ehrenfeucht-Fraïssé argument to establish elementary equivalence. ■

3.6 Bisimulation as a game

Bisimulation can naturally be thought of as a form of process equivalence, but a more dynamic perspective is also possible. We have already seen that the modal satisfaction definition can be recast in the form of a game (recall Figure 6) but the task of determining whether two models are bisimilar can also be viewed in this way. Consider a game between Spoiler (the difference player) and Duplicator (the similarity player) comparing successive pairs in two pointed models $(\mathfrak{M}, w)$ and $(\mathfrak{N}, w')$:

> *If w and w' do not agree on atomic information, Spoiler wins the game in zero rounds. In subsequent rounds, Spoiler chooses a state in one model which is a successor of the current w or w', and Duplicator responds with a matching successor in the other model. If the chosen points differ in their atomic properties, Spoiler wins. If one player cannot move, the other wins. Duplicator wins on infinite runs on which Spoiler does not win.*

This game captures the zigzag behaviour of bisimulations in an obvious sense. It is also *determined*: one of the two players has a winning strategy. (This is because it is an open Gale-Stewart game in the sense of game theory.) For example, returning yet again to the models $\mathfrak{M}$, $\mathfrak{N}$ and $\mathfrak{K}$ considered at the start of this section, we see that Duplicator has a winning strategy in the comparison game for the models $\mathfrak{M}$ and $\mathfrak{K}$ starting from their matched designated points, while Spoiler has one for $\mathfrak{M}$ and $\mathfrak{N}$. The following result clarifies the role of these games precisely:

LEMMA 17 (Adequacy of Modal Comparison Games).

1. *There is an explicit correspondence between Spoiler's winning strategies in a k-round comparison game between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ and modal formulas of modal operator depth k on which s and t disagree.*

2. *There is an explicit correspondence between Duplicator's winning strategies over an infinite-round comparison game between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ and the set of all bisimulations between $\mathfrak{M}$ and $\mathfrak{N}$ that link the points s and t.*

Proof. This result is essentially a fine-grained restatement of the Lemma 9 from a game-theoretic perspective. See Chapter 5 of this handbook for more on game-based approaches to bisimulation. ■

For example, in the game between the models $\mathfrak{M}$ and $\mathfrak{K}$ given earlier, Duplicator wins by choosing responses that stick to the bisimulation links. And in the game between $\mathfrak{M}$ and $\mathfrak{N}$, Spoiler can win in at most three rounds by using the earlier modal difference formula $\Box(\Box\bot \vee \Diamond\Box\bot)$ of modal operator depth three. In each round he can make sure that some modal difference remains at the current match, with the modal operator depth descending each time.

4 COMPUTATION AND COMPLEXITY

We view modal logic as a tool for representing and reasoning about graphs. Our discussion of expressivity has given us some insight into the representational capabilities of modal logic (at least at the level of models) but what about reasoning?

In this section we discuss modal reasoning from a computational perspective. We concentrate on the *model checking task* and the *satisfiability and validity* problems, but also make some remarks about the *global satisfiability* and the *model comparison* tasks. As we shall see, the complexity of the modal version of these tasks is lower than that of their first-order counterparts.

Before going further, two general remarks. First, although we are about to study reasoning, we are not about to embark on the study of modal proof systems (apart from anything else, the standard proof systems are only relevant to satisfiability and validity checking, and there is more to modal reasoning than this). Secondly, although we are ostensibly moving on from expressivity issues to computational issues, the two topics are intertwined. In essence, the positive computational results reported here arise from negative expressivity results (for example, the inability of the basic modal language to force the existence of infinite models).

4.1 Model checking

The model checking task can be formulated locally:

> *Given a (finite) model $\mathfrak{M}$, a point w in $\mathfrak{M}$, and a basic modal formula φ, is φ satisfied in $\mathfrak{M}$ at w?*

Or globally:

> *Given a (finite) model $\mathfrak{M}$, and a basic modal formula φ, is φ satisfied at all points in $\mathfrak{M}$?*

Or in a form that subsumes both the local and global perspectives:

> *Given a (finite) model $\mathfrak{M}$, and a basic modal formula φ, return the set of points in $\mathfrak{M}$ that satisfy φ.*

In what follows we shall work with the last formulation, which is probably the most common way of thinking about model checking in practice.

Now, model checking is clearly a task with computational content — but is it really a *reasoning* task? In our view, yes. In essence, a model is a 'flat' store of information: it consists of a collection of entities, together with a specification of which entities have which properties, and which entities are related by which atomic relations. A modal formula, on the other hand, is a recursively constructed tree. The embedding of connectives and modalities within one another permits relatively short formulas to make interesting assertions, assertions that go way beyond the mere listing of atomic facts. If we add to these differences the practical observation that in typical applications the formula will be much smaller than the model, we see that model checking is about synchronising two very different forms of information: it tells us whether the abstract information embodied in the formula is implicitly present in the model, and gives us a set of points where this implicit information emerges. Viewed this way, model checking is a quintessential reasoning task.

Moreover, model checking has turned out to be of great practical importance — indeed, one of the more salutary lessons computer science has taught logic is just how important this modest looking form of reasoning actually is. Nowadays the practical importance of modal model checking dwarfs that of determining modal satisfiability or validity (the tasks logicians have traditionally viewed as paramount) as a wide range of practical tasks can be modeled in a computationally natural manner, and efficiently solved, via model checking. A classic example is hardware verification. Even though a computer chip is a concrete object, it gives rise to a natural abstract model, namely the set of all states the chip can be in, and the transitions between them. If a chip is to work satisfactorily, its computational runs (that is, the sequences of states it can follow by making transitions from the initial state) should possess a number of high-level 'emergent' properties: for example, these runs should not enter deadlock situations. If we have a modal language that can express the desired properties (for example, absence of deadlock) then by checking the formula in the model representing the chip we can determine whether the design is satisfactory or not.

So how should we perform model checking? The standard approach is to use a bottom-up *labeling algorithm*. To model check a formula φ we label every point in the model with all the subformulas of φ that are true at that point. We start with the proposition symbols: the valuation tells us where these are true, so we label all the appropriate points. We then label with more complex formulas. The booleans are handled in the obvious way: for example, we label w with $\psi \wedge \theta$ if w is labeled with both ψ and θ. As for the modalities, we label w with $\Diamond\varphi$ if one of its R-successors is labeled with φ, and we label it with $\Box\varphi$ if all of its R-successors are labeled with φ. A precise definition of the algorithm for checking diamond formulas is given in the pseudo-code of Figure 15.

procedure $Check\Diamond(\psi)$
    $T := \{v \mid \psi \in label(v)\}$;
    while $T \neq \emptyset$ **do**
        choose $v \in T$;
        $T := T \setminus \{v\}$;
        for all w **such that** Rwv **do**
            if $\Diamond\psi \notin label(w)$ then
            $label(w) := label(w) \cup \{\Diamond\psi\}$;
            end if ;
        end for all ;
    end while ;
end procedure

Figure 15. Model checking $\Diamond\psi$.

The beauty of this algorithm is that we never need to duplicate work: once a point is labeled as making φ true, that's it. This makes the algorithm run in time polynomial in the size of the input formula and model: the algorithm takes time of the order of

$$\mathit{con}(\varphi) \times \mathit{nodes}(\mathfrak{M}) \times \mathit{nodes}(\mathfrak{M}),$$

where $\mathit{con}(\varphi)$ is the number of connectives in φ, and $\mathit{nodes}(\mathfrak{M})$ is the number of nodes in $\mathfrak{M}$. To see this, note that $\mathit{con}(\varphi)$ tells us how many rounds of labeling we need to perform, one of the $\mathit{nodes}(\mathfrak{M})$ factors is simply the upper bound on the nodes that need to be labeled, while the other is the upper bound on the number of successor nodes that need to be checked.

Thus modal model checking is a computationally tractable task, but this is not the case for first-order logic. In fact, model checking first-order formulas is a PSPACE-complete task (see

Chandra and Merlin [22]). That is, although it is possible to write an algorithm that solves the first-order model checking task using an amount of computer memory that is only polynomial in the size of the input model and formula, the algorithm may require running time that is exponential in the size of the input. The problem, of course, lies with the quantifiers. Assuming that the standard assumptions made in complexity theory are correct, there is no way of adapting the labeling algorithm (or indeed, any other algorithm) to perform first-order model checking in polynomial time.

However the labeling algorithm sketched above *does* adapt to more powerful modal languages, and this is important. As we said above, when model checking we want to state interesting high-level properties of the situation we are modeling, and often the ordinary $\Box$ and $\Diamond$ modalities simply aren't expressive enough. In model checking applications, it is usual to work with tree-like models, namely trees of computational runs. On such models $\Diamond$ is interpreted as "at some immediate successor state". This is natural, to be sure, but somewhat limited. However, by adding the binary Until modality, we gain access to entire *sequences* of successor states:

$\mathfrak{M}, s \models U(\psi, \theta)$ iff there is a t such that sR^*t and $\mathfrak{M}, t \models \psi$,
and for all u such that sR^*u and uR^+t we have $\mathfrak{M}, u \models \theta$.

Here R^* is the reflexive transitive closure of the "immediate successor" transition relation R explored by $\Diamond$, and R^+ is its transitive closure. Thus Until gives us a direct handle on the computational runs that can be followed in the model, and this clearly places interesting expressive power at our disposal. Nowadays the Until modality is a fundamental component of some of the most important model checking formalisms — formalisms such as LTL (Linear Time Temporal Logic) and CTL (Computational Tree Logic). For an introduction to these logics, see Chapter 11 of this handbook, or Clarke, Grumberg and Peled [25].

We shall examine the Until operator and the extra expressivity it offers more closely in Section 6.3. Here we simply want to address the following question: how do we extend the labeling algorithm to handle formulas of the form $U(\psi, \theta)$? Here's the basic idea. First, if any point w is labeled with ψ, label w with $U(\psi, \theta)$. Second, if any point v is labeled with θ and at least one R-successor of v is labeled with $U(\psi, \theta)$, then label v with $U(\psi, \theta)$. It should be clear that these two steps reflect the semantics for Until just given; the pseudo-code given in Figure 16 shows how to make the basic idea precise.

Now for an important point. Throughout the previous discussion we have tacitly assumed that we have some way of representing formulas and finite models that is suitable for computational implementation. It is probably not worth sketching details of how this might be done: nowadays it seems safe to assume that most readers of a technical book on logic have at least a nodding acquaintance with programming (indeed, we suspect that most of our readers would find it straightforward to devise a computational syntax for models and modal languages, and to implement simple programs for working with them).

Nonetheless, such issues cannot be taken lightly. A major factor in the spectacular progress of model checking has been the development of *Binary Decision Diagrams* (BDDs) and *Ordered Binary Decision Diagrams* (OBDDs). BDDs (which are compact representations of boolean expressions) were introduced by Lee [88] and Akers [3], and OBDDs (a more sophisticated form of BDD with fewer representational redundancies) were introduced by Bryant [17]. BDDs were first proposed for model checking by Burch, Clarke, McMillan, Dill, and Hwang [18] and as the title of this paper indicates ("Symbolic model checking: 10^{20} states and beyond") this led to a dramatic increase in the size of the models that could be handled. It is important not to underestimate the gap between the labeling algorithm sketched above, and what it takes to make

```
procedure CheckU(ψ, θ)
    T := {v | ψ ∈ label(v)} ;
    for all v ∈ T do
        label(v) = label(v) ∪ {U(ψ, θ)} ;
    end for all ;
    while T ≠ ∅ do
        choose v ∈ T ;
        T := T \ {v} ;
        for all w such that Rwv do
            if U(ψ, θ) ∉ label(w) and θ ∈ label(w) then
                label(w) := label(w) ∪ {U(ψ, θ)} ;
                T := T ∪ {w} ;
            end if ;
        end for all ;
    end while ;
end procedure
```

Figure 16. Model checking $U(\varphi, \theta)$.

a working model checker handle a large model. Crossing this gap requires a combination of theoretical insight and computational expertise, and an entire research community is devoted to exploring the issues involved.

For a good textbook level introduction to model checking, see Huth and Ryan [72]. This book not only introduces the basic algorithms, it also shows how they can be implemented with the aid of OBDDs. Moreover, it discusses modal checking for the modal μ-calculus (which we introduce in Section 6.7). For a more advanced treatment, see Clarke, Grumberg and Peled [25]. Finally, for an account of model checking via automata-theoretic methods, see Chapter 17 of this handbook.

4.2 Satisfiability and validity: decidability

It is often said that modal logic is decidable. This can be read as shorthand for the following claim: the *validity problem* for the basic modal language (*given a basic modal formula φ, is φ valid?*) is decidable. That is, it is possible (ignoring constraints of time and space) to write a computer program which takes a basic modal formula as input, and halts after a finite number of steps and correctly tells us whether it is valid or not.

The decidability of modal logic can also be viewed as a claim that the *satisfiability problem* for the basic modal language (*given a basic modal formula φ, is φ satisfiable in some model?*) is decidable. That is, it is possible (again, ignoring constraints of time and space) to write a computer program which takes a basic modal formula as input, and halts after a finite number of steps and correctly tells us whether it is satisfiable in some model or not. The validity and satisfiability problems are *dual problems*: a modal formula φ is valid iff $\neg\varphi$ is not satisfiable, hence if we have a method for solving one problem, we have a method for solving the other. In what follows we show that both problems are decidable; we'll talk in terms of satisfiability.

A lot is known about the decidability of satisfiability problems for various logics, so it is not too difficult to establish modal decidability: we can do so by reducing the problem to known results for other logics. Here's an easy example. The satisfiability problem for the *two variable fragment* of first-order logic (that is, the fragment of first-order logic in which every formula contains only two variables) is decidable. Now, every basic modal formula can be translated into a formula in the two-variable fragment. To see this we need simply make a small adjustment to the standard translation ST_x. Whenever we translate a $\Diamond$ or a $\Box$, instead of choosing a completely new variable to quantify over accessible points, we use a second fixed variable (say y). If we later encounter another $\Diamond$ or $\Box$, we flip back to the original variable x, and so on. More precisely, we redefine ST_x so it always uses y to quantify over accessible points, and define a twin translation ST_y which always quantifies using x. Here are the key clauses:

$$\begin{array}{ll} \mathrm{ST}_x(\Diamond\varphi) = \exists y\,(Rxy \wedge \mathrm{ST}_y(\varphi)) & \mathrm{ST}_y(\Diamond\varphi) = \exists x\,(Ryx \wedge \mathrm{ST}_x(\varphi)) \\ \mathrm{ST}_x(\Box\varphi) = \forall y\,(Rxy \rightarrow \mathrm{ST}_y(\varphi)) & \mathrm{ST}_y(\Box\varphi) = \forall x\,(Ryx \rightarrow \mathrm{ST}_x(\varphi)). \end{array}$$

The interleaving of ST_x and ST_y guarantees that for any basic modal formula φ, $\mathrm{ST}_x(\varphi)$ will contain only the two variables x and y, and it should be clear that the modified translation is equivalent to the original one. It follows that the satisfiability problem for the basic modal language must be decidable: to test a modal formula for satisfiability, simply translate it with this new version of the standard translation, and then apply the satisfiability algorithm for the two-variable fragment to the output.

It is pleasant that modal decidability can be established so easily, but the proof isn't particularly instructive. The following semantic argument is somewhat more revealing. We shall show that the basic modal language has the *finite model property*, or to put it another way, that it does not have the expressive strength required to force the existence of infinite models. Needless to say, this is in sharp contrast with first-order logic: even such a simple first-order formula as

$$\forall x \neg Rxx \wedge \forall xyz(Rxy \wedge Ryz \rightarrow Rxz) \wedge \forall x \exists y Rxy$$

has only infinite models. In fact, the basic modal language has a rather strong form of the finite model property. We shall show the following:

THEOREM 18 (Strong Finite Model Property). *Let φ be a basic modal formula. If φ is satisfiable, then it is satisfiable on a finite model containing at most $2^{s(\varphi)}$ points, where $s(\varphi)$ is the number of subformulas of φ.*

The decidability of the modal satisfiability problem follows immediately from this result. If a modal formula φ is satisfiable at all, it is satisfiable on a model containing at most $2^{s(\varphi)}$ points. As there are (up to isomorphism) only finitely many such models, exhaustive (and exhausting!) search through them all will settle the issue of φ's satisfiability.

Just as important as the result is the method we shall use to prove it: *filtrations*. These are a standard item in the modal logician's toolkit, and have been used to prove completeness and decidability results for many different modal systems. The basic idea underlying the method is simplicity itself: given a modal formula φ and a model $\mathfrak{M}$ that satisfies it, we make a finite model $\mathfrak{M}$ by collapsing to a single point all the points within $\mathfrak{M}$ that satisfy the same subformulas of φ. But there is a tricky issue: how should we define the relation on the collapsed points in such a way that φ remains true in the finite model? Let's work through the details and see.

We shall say that a set of basic modal formulas Σ is *subformula closed* if every subformula of every formula in Σ is a member of Σ (that is, if $\varphi \wedge \psi \in \Sigma$ then so are φ and ψ, and if $\neg\varphi \in \Sigma$ then so is φ; and if $\Box\varphi \in \Sigma$, then so is φ, and so on). We now define:

DEFINITION 19 (Filtrations). Let $\mathfrak{M} = (W, R, V)$ be a model, let Σ be a subformula closed set of formulas, and let $\leftrightsquigarrow_\Sigma$ be the equivalence relation on the states of $\mathfrak{M}$ defined as follows:

$$w \leftrightsquigarrow_\Sigma v \text{ iff for all } \varphi \text{ in } \Sigma\text{: } (\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{M}, v \models \varphi).$$

The official notation for the equivalence class of a point w of $\mathfrak{M}$ with respect to $\leftrightsquigarrow_\Sigma$ is $|w|_\Sigma$, but in what follows we'll usually assume that Σ is clear from context and simply write $|w|$.

Let $W_\Sigma = \{|w| \mid w \in W\}$. Suppose $\mathfrak{M}^f_\Sigma$ is any model (W^f, R^f, V^f) such that:

1. $W^f = W_\Sigma$.
2. If Rwv then $R^f|w||v|$.
3. If $R^f|w||v|$ then for all $\Diamond\varphi \in \Sigma$, if $\mathfrak{M}, v \models \varphi$ then $\mathfrak{M}, w \models \Diamond\varphi$.
4. $V^f(p) = \{|w| \mid \mathfrak{M}, w \models p\}$, for all proposition symbols p in Σ.

Then $\mathfrak{M}^f_\Sigma$ is called a filtration of $\mathfrak{M}$ through Σ. In what follows we'll drop the subscripts and write $\mathfrak{M}^f$ instead of $\mathfrak{M}^f_\Sigma$.

Two points should be made about this definition. First, observe $\mathfrak{M}^f$ is a filtration of $\mathfrak{M}$ through a subformula closed set of formulas Σ, then $\mathfrak{M}^f$ contains at most $2^{|\Sigma|}$ nodes, where $|\Sigma|$ is the cardinality of Σ. This should be clear: after all, the points of $\mathfrak{M}^f$ simply are the equivalence classes in W_Σ, and there cannot be more than $2^{|\Sigma|}$ of these. Second, the previous definition does *not* specify an accessibility relation on W_Σ — it only imposes constraints (namely clauses 2 and 3) on the properties a suitable accessibility relation R^f should have. That the constraints imposed are sensible is shown by the following result:

THEOREM 20 (Filtration Theorem). *Let $\mathfrak{M}^f$ $(= (W_\Sigma, R^f, V^f))$ be a filtration of $\mathfrak{M}$ through a subformula closed set of basic modal formulas Σ. Then for all formulas $\sigma \in \Sigma$, and all nodes w in $\mathfrak{M}$, we have $\mathfrak{M}, w \models \sigma$ iff $\mathfrak{M}^f, |w| \models \sigma$.*

Proof. By induction on the structure of formulas. The case for proposition symbols is immediate from the definition of V^f, and because Σ is closed under subformulas, the inductive step for the boolean connectives is clear.

So suppose $\Diamond\sigma \in \Sigma$ and $\mathfrak{M}, w \models \Diamond\sigma$. Then there is a v such that Rwv and $\mathfrak{M}, v \models \sigma$. As $\mathfrak{M}^f$ is a filtration, by the first constraint on R^f (clause 2 of the previous definition) we have that $R^f|w||v|$. As Σ is subformula closed, $\sigma \in \Sigma$, hence by the inductive hypothesis $\mathfrak{M}^f, |v| \models \sigma$. Hence $\mathfrak{M}^f, |w| \models \Diamond\sigma$.

Conversely, suppose $\Diamond\sigma \in \Sigma$ and $\mathfrak{M}^f, |w| \models \Diamond\sigma$. Then there is a state $|v|$ in $\mathfrak{M}^f$ such that $R^f|w||v|$ and $\mathfrak{M}^f, |v| \models \sigma$. As $\sigma \in \Sigma$, by the inductive hypothesis $\mathfrak{M}, v \models \sigma$. Making use of the second constraint on R^f (clause 3 of the previous definition) yields $\mathfrak{M}, w \models \Diamond\sigma$. ■

It only remains to verify that relations satisfying the constraints demanded of R^f actually exist. They do. Define:

1. $R^s|w||v|$ iff $\exists w' \in |w|\ \exists v' \in |v|\ Rw'v'$.
2. $R^l|w||v|$ iff for all formulas $\Diamond\varphi$ in Σ: $\mathfrak{M}, v \models \varphi$ implies $\mathfrak{M}, w \models \Diamond\varphi$.

It is straightforward to show that both relations satisfy the required constraints. Actually, you can show a little more: if R^f is any relation satisfying the above constraints then $R^s \subseteq R^f \subseteq R^l$. For this reason, R^s and R^l are said to give rise to the smallest and largest filtrations respectively.

So we have proved Theorem 18: the basic modal language indeed has the strong finite model property. As we argued above, this in turn establishes the decidability of the basic modal satisfiability problem. Now, as is well known, the satisfiability problem for full first-order logic is undecidable. First-order logic is the classic example of a language where expressivity has been purchased at the expense of decidability. The basic modal language reverses this trade-off.

4.3 Satisfiability and validity: complexity

What do the decidability proofs just given tell us about the computational complexity of the modal satisfiability problem? Only that it can be solved in NEXPTIME (that is, non-deterministic exponential time). This is clear from the filtration proof: to see if φ is decidable, we can non-deterministically choose a model containing at most $2^{s(\varphi)}$ points, and then check whether or not it satisfies φ. As we have seen from our discussion of model checking, the checking takes time polynomial in the size of model; however as the model is exponential in the size of the input formula φ, this is a complex task. The reduction to the satisfiability problem for the two-variable fragment yields the same upper bound, as this problem is NEXPTIME-complete.

But the satisfiability problem for basic modal logic is PSPACE-complete. That is, given a modal formula φ, it is possible to write an algorithm to determine whether or not φ is satisfiable that uses an amount of computer memory that is only polynomial in the size of φ. Now, most complexity theorists believe that PSPACE-complete problems are harder than the satisfiability problem for classical propositional logic (the classic NP-complete problem) but easier than EXPTIME-complete problems, which in turn are believed to be easier than NEXPTIME-complete problems. So, given standard complexity-theoretic assumptions, the modal satisfiability problem is probably easier than our earlier decidability proofs suggest.

How do we design a PSPACE algorithm for modal satisfiability? We cannot give a detailed answer here, but we can point to an expressive weakness of modal logic which should make it plausible that PSPACE algorithms for modal satisfiability exist.

LEMMA 21. *Let $\mathfrak{M} = (W, R, V)$ be a model, let $w \in W$, let n be a natural number, let $S_{n,w}$ be the subset of W containing w and all points in W reachable from w by making at most n R-transitions, and let $\mathfrak{N}$ be the submodel $(S_{n,w}, R|_S, V|_S)$, where $R|_S$ and $V|_S$ are the restrictions of R and V respectively to $S_{n,w}$. Then, for all basic modal formulas φ such that md$(\varphi) \leq n$, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{N}, w \models \varphi$.*

That is, if we take a model $\mathfrak{M}$, and extract a submodel $\mathfrak{N}$ from it by throwing away all points that are more than n steps away from w, then no formula with modal operator depth of at most n can distinguish the two models at w. Modal formulas have shallow vision. And if we combine this lemma with what we have already learned about finite models and bisimulations, we obtain the following:

THEOREM 22. *Every formula φ in the basic modal language is satisfiable in a model based on a finite tree of depth at most md(φ).*

Proof. As model logic has the finite model property, if a modal formula is satisfiable, it is satisfiable on a finite model $\mathfrak{M}$ at some point w. As we remarked in the previous section, it is always possible to unravel a model into an equivalent tree-based model. Now, if we unravel $\mathfrak{M}$ about w, we don't necessarily obtain a finite model, but (as $\mathfrak{M}$ is finite) we do obtain a model based on a tree with a finite branch factor, and this model satisfies φ at its root. If we then chop off all points more than $md(\varphi)$ away from the root we obtain a finite model which (by the previous lemma) satisfies φ at its root. ■

So every modal formula is satisfiable on a shallow tree, and we are now in a position to appreciate how PSPACE algorithms for modal satisfiability work. In essence, they construct shallow trees branch by branch. If a branch is successfully constructed (something which takes only space polynomial in the size of the input formula, as the length of the branch is bounded by $md(\varphi)$) the branch is discarded (thus freeing up the memory) and the next branch is then constructed. There may be many branches, so it may take exponential time to construct them all, but as all branches are discarded once they are constructed, such an algorithm uses only polynomial space. This sketch has neglected some important issues (such algorithms require space for recording book-keeping details, and we need to ensure that the space used for this is not excessive) but it does describe, in broad terms, how many modal satisfiability algorithms (notably those based on tableaux or games) work.

But we should issue a word of warning: it's not always so easy. Yes, matters are relatively straightforward here, but that is because we have been working with the *basic* modal language over the class of *all* models. If we impose restrictions on the class of models we are working with (as we shall do in Section 5) or work with richer modal languages (as we shall do in Section 6), or both, we can easily find ourselves faced with undecidable, or even highly undecidable, satisfiability and validity problems.

4.4 Other reasoning tasks

We have discussed the big three (model checking, and satisfiability and validity checking) but this by no means exhausts the reasoning tasks of interest. To conclude this section, let's briefly consider two others.

Although we have stressed the locality of modal logic, some problems demand a global perspective. In particular, if we view a modal formula as a general background *constraint*, we will typically want it to be globally satisfied: that is, we will be interested in models $\mathfrak{M}$ such that $\mathfrak{M} \models \varphi$. The importance of the global satisfiability problem has been strongly emphasised by the description logic community. Indeed, description logic builds into its architecture the idea of a *Terminological Box* (or *TBox*), a collection of formulas that encode background knowledge about some domain (for example, that all men are mortal, that all Martians own flying saucers, or that each employee has a social security number). Description logicians are interested in models in which the TBox is globally satisfied, for these are the models that reflect all the background assumptions.

Once the importance of background constraints is realised, it becomes clear that it is not the pure global satisfiability task itself that is of primary interest. Rather, it is the *local-global satisfiability task*: given formulas φ and ψ, is there a model which locally satisfies φ and globally satisfies ψ? That is, is it possible to satisfy φ subject to the global constraint ψ?

Here's an example. Suppose we're working in a zoological setting, and are interested in the interaction of maternal love and professional responsibility on the feeding of our furry ursine brethren. To put it another way, suppose we have the following TBox:

$$\begin{array}{rl} \textit{bear} \vee \textit{human} & \qquad \textit{bear} \rightarrow \langle \textsc{mother} \rangle \textit{bear} \\ \textit{bear} \rightarrow \neg \textit{human} & \qquad \textit{bear} \rightarrow [\textsc{fedby}](\textit{zoo-keeper} \vee \textit{mother}) \end{array}$$

Let's call this TBox BEAR-CARE. The sort of queries we might be interested in posing are: is it possible to globally satisfy BEAR-CARE and, simultaneously, to locally satisfy

$$\langle \text{MOTHER} \rangle (\textit{bear} \wedge \textit{human})?$$

(No, it's not.) And is it possible to globally satisfy BEAR-CARE and simultaneously to locally satisfy

$$\langle \text{FEDBY} \rangle (\neg human \wedge \neg \textit{mother})?$$

(Yes, it is: BEAR-CARE doesn't rule out having bears as zoo-keepers. This may well be a bug in the TBox.)

Local-global satisfiability problems are also natural in the setting of parsing problems. It is possible to encode various kinds of grammars (such as regular grammars or context-free grammars) as modal formulas (see Chapter 19 of this handbook for a discussion of such approaches). Then, given a string of symbols, the parsing problem is to decide whether it is possible to find a model which embodies all the constraints encoded in the grammar, and which simultaneously satisfies the formula encoding the input string. That is, we would like to globally satisfy the modal formula GRAMMAR and simultaneously locally satisfy INPUT-STRING.

Unsurprisingly, both the global, and the local-global satisfiability tasks are tougher than the ordinary satisfiability problem:

THEOREM 23. *The global satisfiability and the local-global satisfiability tasks for basic modal languages are both EXPTIME-complete.*

Proof. The stated result is an immediate consequence of Hemaspaandra's [118, 65] complexity results for the universal modality (we introduce the universal modality in Section 6.1). But the result holds for even stronger languages; see De Giacomo and Lenzerini [28] for related results for more expressive description logics. ■

EXPTIME-complete problems are decidable but provably intractable: they contain problem instances that will require time exponential in the size of the input to solve (which can mean that they require more time than the expected lifetime of the universe). This, however, is a worst-case scenario. One of the most important recent developments in computational logic has come from the description logic community, who have shown it is possible to specify and implement tableaux-based algorithms for such problems that are remarkably efficient in practice. Moreover, interesting work exists on performing modal theorem proving via (non-standard) translations into first-order logic, so that optimised first-order resolution provers can be applied to the task. For a detailed discussion and comparison of these methods, see Chapter 4 of this handbook, and for a deeper examination of the complexity of modal logic, see Chapter 3.

We conclude with a remark on the *model comparison* task. As bisimulation is the modally fundamental notion of graph equivalence, it is natural to wonder how difficult it is to determine when two models are bisimilar. The corresponding problems for first-order logic (namely, testing for graph isomorphism) is thought to be difficult: there is no known polynomial algorithm for testing for graph isomorphism, though the problem has not been shown to be NP-complete either. In fact, the problem of identifying isomorphic graphs is sometimes regarded as giving rise to a special complexity class of its own.

Testing for bisimulation, however, turns out to be computationally tractable, and there are elegant polynomial algorithms which work by discarding pairs of point that cannot make it into any bisimulation (see Dovier, Piazza and Policriti [33]). Again an expressivity result lies behind this result: the maximal bisimulation between two models $\mathfrak{M}$ and $\mathfrak{N}$ is explicitly definable in a first-order fixed-point language over the disjoint union $\mathfrak{M} \uplus \mathfrak{N}$ of the two models. Such languages have been studied extensively in computer science, and they are known to have good computational behaviour.

Let us summarise our discussion. For a number of tasks, the basic modal language (interpreted over the class of all models) is computationally better behaved than the corresponding first-order language (interpreted over the same models). Figure 17 summarises the relevant facts (PTIME is short for Polynomial Time). Of course, this better computational behaviour comes about because

	Model Checking	Satisfiability	Model Comparison
FOL	PSPACE-complete	Undecidable	in NP
ML	PTIME	PSPACE-complete	PTIME

Figure 17. First-order logic and modal logic: computational properties summarised.

the basic modal language is not nearly as expressive as first-order logic. Thus the pressing questions are: what are the trade-offs? And can this better computational behaviour be lifted to more expressive modal logics, and (if so) how? We shall revisit these questions in the following two sections.

5 RICHER LOGICS

Until now, we have deliberately said rather little about modal *logics* and what they are. Instead we have acted as if there was only one modal logic of any interest, namely the set of valid formulas (that is, the set of formulas satisfied at all points in all models) or, to put it syntactically, the set of formulas generated by the minimal proof system **K** (which we defined at the start of Section 2.2). But traditional presentations of modal logic tend to emphasise the *multiplicity* of modal logics, and devote a great deal of attention to logics richer than **K**, logics with such names as **T**, **K4**, **S4**, **S5**, **GL**, and **Grz**. Where do richer modal logics come from?

As a first approximation (we'll shortly see why it's only an approximation) we might say that richer logics emerge at the level of *frames*, via the concept of *frame validity*. Let $\varphi(p_1, \ldots, p_n)$ be a basic modal formula built out of the proposition symbols $p_1, \ldots, p_n$. We say that $\varphi(p_1, \ldots, p_n)$ is *valid on a frame* $\mathfrak{F} = (W, R)$ *at a point* w if, for each valuation V for its proposition symbols $p_1,\ldots,p_n$, we have that φ is satisfied in the resulting model at w; in such a case we write $\mathfrak{F}, w \models \varphi$. We say φ is *valid on* $\mathfrak{F}$ if it is valid at each point in $\mathfrak{F}$, and we write this as $\mathfrak{F} \models \varphi$. Moreover, we say that a modal formula is *valid on a class of frames* F if it is valid on each frame $\mathfrak{F}$ in F. Note that a valid formula (as defined in Section 2.1) is simply a formula that is valid on the class of all frames.

The starting point for this section is the observation that different applications of modal logic typically validate different modal axioms, axioms over and above those to be found in the minimal system **K**. For example, if we view our models as flows of time, it is natural to assume that the accessibility relation is transitive, and (as the reader should check) any instance of the schema $\Box\varphi \rightarrow \Box\Box\varphi$ is valid on the class of transitive frames (for example, the formula $\Box p \rightarrow \Box\Box p$ is valid on such frames, and $\Box(p \vee q) \rightarrow \Box\Box(p \vee q)$ is too). However no instance of this schema (which for historical reasons is called 4) is provable in **K**, so if we want a logic for working with temporal flows we should add all its instances as extra axioms, and doing so yields the logic known as **K4**. Or suppose we are modeling situations where the frame relation has to be treated as a partial function. As the reader should check, all instances of the schema $\Diamond\varphi \rightarrow \Box\varphi$ are valid on the class of such frames, and none of them can be proved in **K**, so once again we should add them as extra axioms. Doing so yields the logic called $\mathbf{KAlt}_1$.

We begin this section by briefly discussing such axiomatic extensions of **K** a little further. But our real interest is not the richer logics that arise by adding extra axioms (for an introduction to this topic, see Chapter 2 of this handbook) rather it centres on the following semantic questions: what can modal formulas say about frames, and how do they say it? As we shall see, there is a fundamental expressivity distinction between the level of models and the level of frames: whereas modal logic at the level of models is the bisimulation invariant fragment of first-order logic, at the level of frames it is a fragment of second-order logic.

5.1 Axioms and relational frame properties

One of the most attractive features of modal logic is the illumination provided by the fact that modal axioms reflect properties of accessibility relations. A typical modal completeness theorem reads like this:

THEOREM 24. *A formula is provable in **S4** iff it is true in all models based on frames whose accessibility relation is transitive and reflexive.*

Proof. See Chapter 2 of this handbook (or indeed, virtually any introduction to modal logic). ■

That is, the theorems of **S4** are true in all graphs with a transitive and reflexive relation, while its non-theorems have some transitive and reflexive counter-model; the additional axioms reflect simple visualisable geometric conditions in the semantics. There are many techniques for proving such completeness results, ranging from simple inspection of the *canonical model* constructed from all complete theories in the logic (this fundamental technique is introduced in Chapter 2 of this handbook) to various types of model surgery (such as filtration, unraveling, and taking bounded morphic images). Moreover, the motivations for proving modal completeness theorems may differ. Sometimes we start with an independently interesting proof system and try to find a useful corresponding class of frames. The classic example of this is the proof system **GL**, that is **K** enriched with all instances of the Löb axiom schema $\Box(\Box\varphi \rightarrow \varphi) \rightarrow \Box\varphi$, which arose via the study of arithmetical provability (see Chapters 2 and 16 of this handbook for further discussion of **GL**) and was later proved complete with respect to the class of finite trees (where the binary relation interpreting the modalities is the transitive closure of the one-step daughter-of tree relation). Sometimes, however, we might start with a natural model class — say an interesting space-time structure — and try to axiomatise its modal validities. The literature is replete with both variants.

Nowadays a lot is known about axiomatic extensions of **K**. For a start, it turns out that there are uncountably many such *normal modal logics*, as they are often called. It is usual to identify a normal modal logic with the set of formulas it generates, and we say that a modal logic is consistent if it does not contain all formulas. This identification immediately induces a lattice structure on the set of all such logics. The cartography of this landscape is an object of study in its own right; here we shall only mention that, because of the following result, it contains two major highways.

THEOREM 25. *Let **Id** be the normal modal logic generated by **K** enriched with all instances of the axiom schema $\varphi \leftrightarrow \Box\varphi$, and let **Un** be the normal modal logic generated by **K** enriched with the axiom $\Box\bot$. Every consistent normal modal logic is either a subset **Id** or **Un**.*

Proof. See Makinson [92] for the original (algebraic) proof. After we have introduced generated submodels and bounded morphisms for frames we will be able to sketch the semantic ideas that underly this result, and we shall do this shortly. ■

Now, as the reader should check, every instance of $\varphi \leftrightarrow \Box\varphi$ is valid on frames which consist of a collection of isolated reflexive points, and $\Box\bot$ is valid on frames consisting of a collection of isolated irreflexive points. Moreover, using standard techniques it is easy to show that **Un** is complete with respect to the first frame class, and **Id** with respect to the second. Thus the semantic content of Theorem 25 is that every normal modal logic is contained in the logic of one of these frame classes; for example, **S4** lies on the first road, and **GL** on the second.

But the most important fact to have emerged about normal modal logics is that *not* all of them have frame-based characterisations. In fact, frame completeness results (such as the result for **S4** noted above) are the exception rather than the rule. Thus our earlier remark that richer logics emerged at the level of frames via the concept of frame validity was very much a first approximation: the notion of frame validity simply does not provide an adequate semantic basis for studying all normal modal logics. Here is a concrete example of a *frame incompleteness* result:

THEOREM 26. *Let **TMEQ** be the normal modal logic obtained by enriching **K** with all instances of the following schemas:* $\varphi \rightarrow \Diamond\varphi$ ***(T)**,* $\Box\Diamond\varphi \rightarrow \Diamond\Box\varphi$ ***(M)**,* $\Diamond(\Diamond\varphi \wedge \Box\psi) \rightarrow \Box(\Diamond\varphi \vee \Box\varphi)$ ***(E)**, and* $(\Diamond\varphi \wedge \Box(\varphi \rightarrow \Box\varphi)) \rightarrow \varphi$ ***(Q)**. There is no class of frames that validates precisely the formulas in **TMEQ**.*

Proof. See van Benthem [129]. ■

Such incompleteness results (which were first proved in the early 1970s by Thomason [125] and Fine [43]) were important in the development of modal logic. For a start, they forced modal logicians to examine alternative ways of semantically characterising normal modal logics, and this led to a renaissance in algebraic semantics of modal logic (see Chapter 6 of this handbook for more on this topic). But they also had another effect, one more relevant to the present chapter: they stimulated a wave of semantic research at the level of frames. This new wave of research was centred around the notion of frame definability, the topic to which we now turn.

5.2 Frame definability and undefinability

Before getting to work, a brief remark. There is another way of thinking about axiomatic extensions of **K**. Instead of viewing them as giving rise to brand new modal logics, we can simply view them as *theories* constructed over the minimal logic **K** in much the same way as a first-order theory (of say, linear orders) is constructed over the set of first-order validities. Nothing of substance hangs on this shift of perspective, but it fits more naturally with our focus on expressivity.

So, bearing this in mind, let's pose the first question: what can modal formulas say about frames? A natural way to approach this is to introduce the concept of *frame definability*. We shall say that a modal formula φ defines a class of frames F iff it is valid on precisely the frames in F. That is, not only must φ be valid on every frame in F, it must also be possible to falsify φ on any frame that is not in F. So, what classes of frames can modal languages define? Here are some simple examples:

PROPOSITION 27.

1. $\Box p \rightarrow \Box\Box p$ *defines the class of transitive frames; that is, frames such that* $\forall xyz(Rxy \wedge Ryz \rightarrow Rxz)$.

2. *$\Diamond p \rightarrow \Box p$ defines the class of frames where the frame relation R is a partial function; that is, frames such that $\forall xyz(Rxy \wedge Rxz \rightarrow y = z)$.*

3. *$p \leftrightarrow \Box p$ defines the class of frames which consist of isolated reflexive points; that is, frames such that $\forall xy(Rxy \leftrightarrow x = y)$.*

4. *$\Box \bot$ defines the class of frames which consist of isolated irreflexive points; that is, frames such that $\forall xy \neg Rxy$.*

Proof. We have already asked the reader to check that these formulas are valid on the class of frames in question. So to complete the proofs of these definability claims we need merely check that each formula can be falsified on any frame that does not belong to the relevant class.

Let's deal with the second example. Suppose (W, R) is a frame where R is not a partial function. This means that there is a point $w \in W$ that has two distinct R-successors, say u and v. It follows that we can falsify $\Diamond p \rightarrow \Box p$ on (W, R) at w. For let V be the valuation that makes p true at u and nowhere else. Then $(W, R, V), w \models \Diamond p$ but $(W, R, V), w \not\models \Box p$, since p is not true at v. So we have falsified $\Diamond p \rightarrow \Box p$ on (W, R) as required. ■

A remark on terminology. Instead of saying, for example, that $\Box p \rightarrow \Box\Box p$ defines the class of transitive frames, we often simply say that $\Box p \rightarrow \Box\Box p$ defines transitivity. It is also usual to say that $\Box p \rightarrow \Box\Box p$ corresponds (at the level of frames) to $\forall xyz(Rxy \wedge Ryz \rightarrow Rxz)$, or that $\forall xyz(Rxy \wedge Ryz \rightarrow Rxz)$ is a frame correspondent for $\Box p \rightarrow \Box\Box p$.

Now for an important question: how do we go about showing that a class of frames *cannot* be modally defined? Answering such questions is typically more demanding than proving the type of result noted in Proposition 27, for instead of checking that a given formula defines a given frame class, we now have to prove that no modal formula is capable of this. How can we prove such general results? By finding ways of transforming frames that preserve frame validity. For if we can show that a class of frames F is *not* closed under such a transformation, it follows that F is *not* modally definable. Let's take a closer look.

The first step is to find transformations that preserve frame validity. Three lie close to hand: the formation of disjoint unions, generated submodels, and bounded morphic images. In Section 3.2 we defined these constructions at the level of models, and they can be lifted to the level of frames simply by ignoring the requirements imposed on the valuations. For example, a bounded morphism between frames (W, R) and (W', R') is a function f from W to W' that satisfies the morphism condition (if Rwv, then $R'f(w)f(v)$) and the zag condition (if $R'f(w)v'$, then there exists a v such that $f(v) = v'$ and Rwv), and we say that frame (W', R') is a bounded morphic image of frame (W, R) if there is a surjective bounded morphism from (W, R) to (W', R'). Lifting these constructions to the level of frames immediately gives us three validity preservation results:

THEOREM 28. *For all basic modal formulas φ we have that:*

1. *Let $\{\mathfrak{F}_i \mid i \in I\}$ be a family of frames. Then if $\mathfrak{F}_i \models \varphi$ for every i in I, we have that $\biguplus \mathfrak{F}_i \models \varphi$ too. That is, frame validity is preserved under the formation of disjoint unions.*

2. *Let $\mathfrak{F}'$ be a generated subframe of $\mathfrak{F}$. Then if $\mathfrak{F} \models \varphi$, we have that $\mathfrak{F}' \models \varphi$ too. That is, frame validity is preserved under the formation of generated subframes.*

3. *Let $\mathfrak{F}$ and $\mathfrak{F}'$ be frames and f a surjective bounded morphism from $\mathfrak{F}$ to $\mathfrak{F}'$. Then if $\mathfrak{F} \models \varphi$, we have that $\mathfrak{F}' \models \varphi$ too. That is, frame validity is preserved under the formation of bounded morphic images.*

Proof. We prove the result for bounded morphisms; we show the contrapositive. Given frames $\mathfrak{F} = (W, R)$ and $\mathfrak{F}' = (W', R')$ such that $\mathfrak{F}'$ is a bounded morphic image of $\mathfrak{F}$ under f, suppose that $\mathfrak{F}' \not\models \varphi$. This means that for some valuation V' on $\mathfrak{F}'$ and some point $w' \in W'$ we have that $(\mathfrak{F}', V'), w' \not\models \varphi$. Let V be the valuation on $\mathfrak{F}$ defined by $V(p) = \{u \in W \mid f(u) \in V'(p)\}$, for all proposition symbols p. Furthermore, let w be any point such that $f(w) = w'$; there must be at least one such point as f is surjective. Then the model $(\mathfrak{F}', V')$ is a bounded morphic image of the model $(\mathfrak{F}, V)$, and hence $(\mathfrak{F}, V), w \not\models \varphi$. ■

Applying this theorem immediately gives rise to a crop of non-definability results. Here are some simple ones. Basic modal languages cannot define the class of simply connected frames, that is, the class of frames such that $\forall xy(Rxy \vee Ryx)$. Why not? Because this class is not closed under the formation of disjoint unions: taking the disjoint union of two frames with this property clearly results in a frame without it. As a second example, the basic modal languages cannot define the class of frames containing an isolated reflexive point. Why not? Because this class is not closed under the formation of generated subframes. For consider a frame consisting of two isolated points, one reflexive, the other irreflexive. This frame belongs to the required class, however the subframe generated by the irreflexive point does not. As a third example, the class of irreflexive frames is not modally definable. Why not? Because it is not closed under the formation of bounded morphic images (recall the bounded morphism of Figure 13 which collapses the natural numbers to a single reflexive point). But frame validity is preserved under this transformation, hence no modal formula can define irreflexivity. For more sophisticated applications of these validity preservation results, see van Benthem [137].

These results also give us insight into the semantic ideas behind Theorem 25. For consider a consistent normal logic. Suppose one of the frames on which it is valid contains an isolated irreflexive point; then (appealing to the preservation of validity under generated subframes) the frame consisting of just that single point validates the logic too. So suppose that no frame containing an isolated point validates the logic. But this means that in all frames that validate the logic, every point has at least one successor. But if we map all the points in such a frame to a singleton reflexive point, the mapping is a bounded morphism. Hence it follows that the logic is validated on frames consisting of isolated reflexive points.

As we shall soon see, the three frame transformations just introduced all play a role in the Goldblatt-Thomason Theorem, a characterisation of modally definable classes of elementary frames. But a fourth transformation, namely the formation of *ultrafilter extensions*, is also needed to complete the statement of this celebrated result, so let's take this opportunity to define this (somewhat more complex) frame construction. First we recall a standard mathematical concept. Given a non-empty set W, a *filter* F over W is any subset of 2^W (the power set of W) that contains W and is closed under finite intersection (that is, if $X, Y \in F$ then $X \cap Y \in F$) and set-theoretic inclusion (that is, if $X \in F$ and $X \subseteq Y \subseteq W$ then $Y \in F$). A filter is called *proper* if it is distinct from 2^W. An *ultrafilter* is a proper filter U such that for all $X \in 2^W$, $X \notin U$ iff $(W \backslash X) \in U$. A standard result assures us that any proper filter can be extended to an ultrafilter. Bearing this in mind, we make the following definition:

DEFINITION 29 (Ultrafilter Extensions of Frames). Let $\mathfrak{F} = (W, R)$ be a frame. For any $X \subseteq W$ we define $l(X)$ to be $\{w \in W \mid \text{for all } v \in W, \text{if } Rwv \text{ then } v \in X\}$. Then the ultrafilter extension *ue*($\mathfrak{F}$) of $\mathfrak{F}$ is defined to be the frame $(\mathit{uf}(W), R^{ue})$, where *uf*(W) is the set of all ultrafilters on W and R^{ue} is the relation consisting of all pairs of ultrafilters U, U' such that for all $X \subseteq W$, if $l(X) \in U$, then $X \in U'$.

We can now state the required theorem. Note that the direction of validity preservation is the reverse of that found in Theorem 28. That is, here frame validity is preserved from the transformed frame (here the ultrafilter extension) back to the original one:

THEOREM 30. *For any basic modal formula φ, if ue$(\mathfrak{F}) \models \varphi$ then $\mathfrak{F} \models \varphi$ does too. That is, frame validity reflects ultrafilter extensions.*

Proof. The use of ultrafilter extensions in modal logic traces back to Goldblatt [57, 58], van Benthem [130], and Fine [44]. For a detailed proof of this theorem, see Proposition 2.59 and Corollary 3.16 of Blackburn, de Rijke and Venema [13]. ■

Although this transformation is harder to visualise than the previous three, it too gives rise to some simple non-definability results. Here's a nice example, taken from Goldblatt and Thomason [60], showing that the class of frames satisfying $\forall x \exists y(Rxy \wedge Ryy)$ is not modally definable. We can see this as follows. The ultrafilter extension of $(\mathbb{N}, <)$, the natural numbers in their usual order, looks a bit like a gigantic lolly-pop. It has an infinite handle, an isomorphic copy of $(\mathbb{N}, <)$, consisting of all the principal ultrafilters (that is, those ultrafilters which contain a singleton set $\{n\}$, where n is a natural number). This is followed by the lolly: an uncountable collection of non-principal ultrafilters which are all related to one another and reflexively related to themselves. Hence $ue(\mathbb{N}, <)$ has the property $\forall x \exists y(Rxy \wedge Ryy)$. Why? Because every point in the frame is related to the reflexive points in the lolly. However this formula is clearly not valid on the original frame $(\mathbb{N}, <)$. As frame validity reflects ultrafilter extensions, it follows that the class of frames satisfying $\forall x \exists y(Rxy \wedge Ryy)$ is not modally definable. For further discussion of ultrafilter extensions from a model-theoretic perspective, see Chapter 5 of this handbook. There is also an important algebraic perspective on ultrafilter extensions, which is discussed in Chapter 6.

5.3 Frame correspondence and second-order logic

Now that we have some idea of what basic modal languages can (and cannot) say about frames, we turn to the second question: how do they say it? And here we encounter something interesting. Note that all four classes of frames mentioned in Proposition 27 are definable by simple first-order formulas — and this is actually rather puzzling. After all, if you think about what it means for a basic modal formula $\varphi(p_1, \ldots, p_n)$ to be valid on a frame, we see that this concept is essentially *second-order*: we quantify across all possible valuations, and valuations assign *subsets* of frames to proposition symbols.

We can make this second-order perspective precise with the help of the standard translation. Let $\mathfrak{F}$ be a frame, let $\mathfrak{M} = (\mathfrak{F}, V)$ be any model over $\mathfrak{F}$, and let w be any point in $\mathfrak{F}$. By Proposition 3 we have that

$$(\mathfrak{F}, V), w \models \varphi(p_1, \ldots, p_n) \text{ iff } (\mathfrak{F}, V) \models \mathrm{ST}_x(\varphi)(P_1, \ldots, P_n)[x \leftarrow w].$$

(Here $P_1, \ldots, P_n$ are the monadic predicate symbols used to translate the proposition symbols $p_1, \ldots, p_n$.) How do we lift this equivalence (which lives at the level of models) to an equivalence at the level of frames (the level where validity is the primary semantic concept)? Very straightforwardly. A formula is valid on a frame iff it is satisfied at any point in the frame under any assignment of subsets of the frame to the proposition symbols. So we only need to universal quantify over the points that can be assigned to x (a first-order quantification) and over the assignments to the monadic symbols $P_1, \ldots, P_n$ (a second-order quantification). Doing so gives us the fundamental correspondence between frame validity and second-order logic:

$$\mathfrak{F} \models \varphi(p_1, \ldots, p_n) \text{ iff } \mathfrak{F} \models \forall P_1 \cdots P_n \forall x \mathrm{ST}_x(\varphi).$$

In short, frame validity systematically treats modal formulas φ as the universal monadic second-order closure of their standard first-order translations on relational models. The second-order upgrade of the first-order correspondence language is often called the *frame correspondence language* or the *second-order correspondence language*.

Let's look at an example. Recall that in Section 2.2 we showed that the standard translation of $p \to \Diamond p$ was $Px \to \exists y(Rxy \wedge Py)$. So if we ask what $p \to \Diamond p$ defines at the level of frames we can give an immediate answer: it defines the class of frames satisfying the following monadic second-order formula:

$$\forall P \forall x(Px \to \exists y(Rxy \wedge Py)).$$

Now, it's certainly pleasant to be able to systematically calculate frame correspondences for modal formulas in this way — but the puzzle remains. Indeed, if anything it has become more acute. For most of the modal formulas encountered in practice correspond to simple first-order conditions on frames, yet these conditions are systematically expressed using rather complex second-order expressions. The translation just considered is a good example. As the reader should check, $p \to \Diamond p$ corresponds to the first-order formula $\forall x Rxx$ (that is, it defines reflexivity). And if you think about it, you will see that $\forall P \forall x(Px \to \exists y(Rxy \wedge Py))$ is indeed a rather roundabout way of expressing reflexivity. For a start, it's easy to see that this sentence is true on any reflexive frame. Conversely, if this sentence is true on a frame (W, R), then $Px \to \exists y(Rxy \wedge Py)$ must be true under any assignment to the free variables x and P. Hence, for any $w \in W$, this formula is true if we assign w to x and $\{w\}$ to P. This assignment makes the antecedent true (indeed, it is the *minimal* valuation required to make the antecedent true; the significance of this remark will become clear when we discuss the Sahlqvist Correspondence Theorem) so we must have that $\exists y(Rxy \wedge Py)$ is true too. But this is only possible if Rww. Hence, as w was arbitrary, this means that R must be reflexive, and thus the original second-order sentence really does express reflexivity. As we said earlier, one of the key questions we are interested in is *how* modal languages talk about frames. And now we have an answer. They do so via a detour through second-order logic.

Moreover, this detour is *not* eliminable. That is, while experience shows that most common modal formulas correspond to first-order conditions on frames, some modal formulas define conditions that are *not* elementary. A famous case is Löb's formula, $\Box(\Box p \to p) \to \Box p$, the characteristic axiom of the logic **GL**. This defines the conjunction of the transitivity of R with the converse well-foundedness of R (that is, it forbids the existence of infinite chains of related points $w_1 R w_2 R w_3 R w_4 R w_5 \ldots$). This condition is non-elementary, as an appeal to the Compactness Theorem for first-order logic shows. Another well-known modal axiom that defines a non-elementary class of frames is the McKinsey formula $\Box\Diamond p \to \Diamond\Box p$. This can be shown by appealing to the Löwenheim-Skolem Theorem for first-order logic. For full proof details for both the Löb and McKinsey examples, see Blackburn, de Rijke and Venema [13].

Summing up, we are confronted with an intriguing situation. At the level of frames, modal formulas systematically correspond to second-order conditions on frames. Nonetheless, in many common cases these second-order conditions turn out to be equivalent to first-order conditions. This raises some interesting questions. Are there criteria that demarcate modal formulas that are essentially first-order at the level of frames from the genuinely second-order ones? And can we characterise the elementary frame classes that are modally definable?

5.4 *First-order frame definability*

As we have just learned, the link between first-order definable frame classes and modal logic is not straightforward. Nonetheless, some elegant general results are known, and we shall briefly discuss three of them here. We first note two results which bear upon the demarcation issue: the Sahlqvist Correspondence Theorem (which isolates a large class of formulas all of which define elementary classes of frames) and a model-theoretic characterisation of the modal formulas which define elementary frame classes. Following this we discuss the celebrated Goldblatt-Thomason Theorem, a model-theoretic characterisation of the elementary frame classes that are basic modal definable. All three results (and others bearing on the theme of elementary frame definability) are discussed in greater detail in Chapter 5 of this handbook.

Let's start with the Sahlqvist [111] result. Upon closer inspection, first-order frame conditions often arise because of the syntactic shape of the defining modal formula — for example the quantifier shape of the first-order formula for transitivity is matched by the sequence of boxes in $\Box p \to \Box\Box p$. The following theorem gives us a natural account of such correspondences. It trades systematically on the idea (noted when we discussed the second-order definition of reflexivity) of substituting minimal verifying valuations in antecedents.

THEOREM 31 (Sahlqvist Correspondence Theorem). *There is an effective method for computing first-order equivalents for Sahlqvist formulas, that is, formulas of the form $\varphi \to \psi$ with antecedents φ constructed from atoms (possibly prefixed by boxes) using conjunctions, disjunctions and diamonds, while consequents ψ can be any modal formula with only positive occurrences of proposition symbols.*

Proof. The effective method (in the form originally introduced by van Benthem [128, 131]) is usually called the substitution algorithm. The following example will give an idea of how it works. The 4 formula, $\Box p \to \Box\Box p$, is a Sahlqvist formula and its second-order translation is

$$\forall P \forall x(\forall y(Rxy \to Py) \to \forall y(Rxy \to \forall z(Ryz \to Pz))).$$

Now, if we could eliminate all the occurrences of P in this formula, we would render the second-order quantification needed to express validity vacuous. But can P be eliminated in a semantically sensible way? Because of the syntactic restrictions that Sahlqvist formulas conform to, it turns out that it can. We do so by replacing P by a first-order expression describing the *minimal* valuation needed to make the antecedent of $\Box p \to \Box\Box p$ true. Now, the minimal way of making $\Box p$ true is to make p true at all successors of the point of evaluation x, so the required substitution is $Pu := Rxu$. Performing this substitution yields the following first-order expression:

$$\forall x(\forall y(Rxy \to Rxy) \to \forall y(Rxy \to \forall z(Ryz \to Rxz))).$$

The antecedent is now tautologically true, and dropping it leaves us with the expression

$$\forall x \forall y(Rxy \to \forall z(Ryz \to Rxz)).$$

But this is a first-order formula expressing transitivity. For a precise specification of the substitution algorithm, and a proof that it works as required, see Blackburn, de Rijke and Venema [13]. The heart of the proof is to show that a Sahlqvist antecedent is true under any value for its proposition symbols iff it is true under its *minimal* values. ■

The Sahlqvist Correspondence Theorem and its proof method are very powerful and can be extended to far stronger modal languages. Nevertheless there are also modal formulas which express first-order conditions on frames that are not covered by the theorem. The **K4.1** axiom

$$(\Box p \rightarrow \Box\Box p) \wedge (\Box\Diamond p \rightarrow \Diamond\Box p)$$

is a conjunction of the 4 axiom with the McKinsey axiom. It defines the class of frames with a transitive and atomic relation, that is the class of transitive frames such that $\forall x \exists y(Rxy \wedge \forall z(Ryz \rightarrow z = y))$. But this first-order equivalence cannot be computed using the substitution method. See van Benthem [137] or Blackburn, de Rijke and Venema [13] for further discussion.

So the Sahlqvist result doesn't fully pin down the modal formulas that define elementary frame classes. However model-theoretic characterisations exist. For example we have:

THEOREM 32. *A modal formula defines a first-order frame property iff it is preserved under taking ultrapowers of frames.*

Proof. For the original proof, see van Benthem [131]. For an introduction to ultrapowers, consult Chang and Keisler [23]. ■

Closure under ultrapowers is an abstract feature, and it is not easy to use it to recognise whether a given modal formula is first-order over frames. But then no simple method can be expected to work: Chagrova [21] shows that the problem of determining whether a modal formula expresses a first-order condition on frames is undecidable.

But now for our other question: which elementary classes of frames are modally definable? The classic result here is the Goldblatt-Thomason Theorem. This tells us that the four frame preservation results noted earlier are not merely necessary, they are also *sufficient* to characterise first-order frame definability:

THEOREM 33 (Goldblatt-Thomason Theorem). *A first-order frame property is modally definable iff it is preserved under taking disjoint unions, generated subframes, bounded morphic images, and reflects ultrafilter extensions.*

Proof. The left-to-right direction is just a restatement of the results noted in Theorems 28 and 30. The real work lies in the converse. The original proof, due to Goldblatt and Thomason [60] was algebraic; we briefly discuss this approach in Section 7.1, and an algebraic proof is given in Chapter 6 of this handbook. Nowadays there are also purely model-theoretic proofs; see van Benthem [133] for the earliest of these. ■

5.5 Correspondence in richer languages

Throughout this section we have kept our eyes firmly on the goal of understanding modal expressivity with respect to elementary frame classes. This is an important topic (after all, we want to understand as much as possible about the route modal logic over frames takes from monadic second-order logic back to first-order logic) but it is also natural to wonder about the expressivity of modal logic with respect to non-elementary frame classes. Unfortunately, it is harder to come up with elegant answers here. In particular, we can't expect sweeping model-theoretic characterisations. Model-theoretic characterisations of elementary frame definability, such as Theorem 32 and the Goldblatt-Thomason Theorem, rest on the conceptual edifice of first-order model theory. Second-order model theory is nowhere near as well developed.

Nonetheless, some interesting results are known. For example, it turns out that we can apply the ideas underlying the proof of the Sahlqvist Correspondence Theorem beyond the confines of first-order logic. Let's briefly consider what is involved. The following discussion is based on van Benthem [138]. Chapter 5 of this handbook contains a more detailed discussion of related material.

The substitution algorithm for Sahlqvist formulas runs into difficulties with more complex antecedents; a classic example is Löb's formula $\Box(\Box p \to p) \to \Box p$, which defines a non-elementary class of frames. But let's reflect on *why* we compute the minimal antecedent values for Sahlqvist formulas. In fact there are two reasons. Firstly, because Sahlqvist antecedents are true under any value for their proposition symbols iff they are true under their *minimal* values. Secondly, because such minimal predicates are first-order definable. Now, as it happens, the Löb antecedent does not fulfil the first-order definability criterion, but this does not mean that all that can be said is that the Löb's formula is intrinsically second-order — for, as it turns out, there is a smallest semantic value for the predicate P which will make its antecedent true. This is the set of points in the frame obtained by taking the *intersection* of all predicates P validating $\Box(\Box p \to p)$ where p is interpreted as P. Such a set must exist, because the standard translation of the Löb antecedent has a special syntactic form. Call a first-order formula $\varphi(P)$ *intersective* if it has one of the forms:

1. $\forall x(\psi(P,Q,x) \to Px)$, with P occurring only positively in $\psi(P,Q,x)$.
2. $\psi(P,Q)$, with P occurring only negatively in ψ.

It is easy to show that all formulas $\varphi(P)$ of this form have the above-mentioned *intersection property*: if $\varphi(P)$ holds for any predicate P it holds for the intersection of all predicates P satisfying it.

Thus it makes sense to talk about *min*$P.\varphi(P)$, the *minimal* satisfying predicate. Of course, such predicates need not be first-order definable, but it is not hard to show that minimal predicates for intersective first-order formulas are definable in a well-known extension of first-order logic, namely *LFP(FOL)*, first-order logic with monotonic fixed-points (we shall introduce the idea of monotonic fixed-points in more detail when we discuss the modal μ-calculus in Section 6). *LFP(FOL)* has many uses in computer science; it lies between first-order and second-order logic, and retains many useful model-theoretic properties such as invariance for potential isomorphism (see Ebbinghaus and Flum [35] for an introduction to *LFP(FOL)*).

Now, once we have such a minimal value for the antecedent predicates, it can be substituted into the consequent to obtain a frame equivalent just as before — though now, of course, we obtain an equivalent in *LFP(FOL)*. To return to our example, the standard translation of the Löb antecedent $\forall y((Rxy \wedge \forall z(Ryz \to Pz)) \to Py)$ is indeed intersective in the above sense. Therefore, the corresponding frame property of the Löb formula can be computed and (as we would expect) the result is an *LFP(FOL)* formula defining the property of transitivity plus converse well-foundedness. As a second example, consider the axiom of cyclic return:

$$(\Diamond p \wedge \Box(p \to \Box p)) \to p.$$

Again, this is not a Sahlqvist formula. But again, the antecedent is intersective (once we have moved out the modal $\Diamond$ to become a prefixed universal quantifier, as before in the substitution algorithm) and gives rise to a simple fixed-point computation for an equivalent frame property:

> *Every point x with an R-successor y can be reached from y by a finite sequence of successive R-steps.*

We can express this condition in *LFP(FOL)* as follows. First we define the concept of transitive closure:

$$R^+xy \ =_{def}\ \textit{min}S, xy.Rxy \vee \exists z(Rxz \wedge Szy).$$

We can then capture the stated frame condition by insisting that:

$$\forall xy(Rxy \to R^+yx).$$

This is the beginning of a further layering of modal formulas with respect to semantic complexity. For there are also modal formulas with frame equivalents which cannot be expressed in *LFP(FOL)*. One example is the well known axiom in tense logic expressing Dedekind Completeness of linear orders, which is not preserved under the potential isomorphism between the rationals and the reals. And recently, van Benthem and Goranko have shown that the McKinsey formula, whose antecedent is typically non-intersective, does not correspond to any *LFP(FOL)* formula.

We started this chapter by saying that the process interpretation is a fundamental way of viewing modal logic. The present discussion shows that there is a natural link between modal logic and a far more sophisticated logic of processes, namely *LFP(FOL)*. We will return to the process interpretation in Section 6 when we examine Propositional Dynamic Logic and the modal μ-calculus, stronger modal languages which, like *LFP(FOL)*, can express some non-elementary concepts, such as transitive closure.

5.6 Remarks on computability

In Section 4 we contrasted the PSPACE decidability of modal logic with the undecidability of first-order logic. But these results concerned satisfiability and validity on the class of all frames. Suppose we restrict attention to particular classes of frames defined by basic modal formulas. There is no reason to suppose that modal satisfiability and validity problems over such frame classes will always be in PSPACE, or even that they will be decidable. And indeed, in many cases they are not not.

In some cases, restricting attention to a certain class of frames may lower the computational complexity. For example, suppose we restrict attention to the frames defined by $\Diamond p \to \Box p$, that is, the class of frames in which R is a partial function. Then the task of testing basic modal formulas for satisfiability becomes NP-complete, that is, no worse than the satisfiability problem for propositional logic. This is because (as the reader can easily check) if a basic modal formula φ has a model based on a frame in this class, then it not only has a finite model in this class, but a model containing at most $n + 1$ points, where n is the number of modalities in φ. Thus a non-deterministic algorithm which guesses a model, checks that it belongs to the frame class, and verifies that the formula is satisfied on it, runs in time polynomial in the size of φ.

But restricting attention to particular frame classes can easily result in undecidable problems. A recurring theme is the distinction between tree-like and grid-like models. We have already discussed why tree-like models are relevant to modal decidability over the class of all models; here we'll merely add that many more modal decidability results can be proved by appealing to *Rabin's Theorem* (see [107]), which in its simplest form shows that the monadic second-order theory of binary branching trees is decidable. Grid-like models, on the other hand, are (roughly speaking) those that contain regions that look like $\mathbb{N} \times \mathbb{N}$ (the product of the natural numbers with itself) under two orderings: the horizontal ordering (that is, $(j, k)R^h(j + 1, k)$), and the vertical ordering (that is, $(j, k)R^v(j, k+1)$) which together give rise to the characteristic grid-like shape. Now, it is hard to give precise generalisations, but experience shows that while even very strong modal languages tend to be decidable over tree-like models, even quite weak languages can be undecidable over grid-like models; we shall note such an example in Section 6 when we discuss combinations of modal logics. Such undecidability results ultimately trace back to the possibility of encoding the $\mathbb{N} \times \mathbb{N}$ *tiling problem*, which is known to be undecidable. For a detailed account of the tiling problem, and a proof that it is undecidable, see Berger [12]. Here we'll simply say that it is essentially a geometrical puzzle. We are presented with a finite collection of square tile types, of fixed orientation. Each edge of each tile type is coloured. The $\mathbb{N} \times \mathbb{N}$ tiling problem asks: is it possible to write an algorithm which, when presented with such a collection of tile types, can correctly determine whether or not $\mathbb{N} \times \mathbb{N}$ can be tiled, using only tiles of the given type, in such a way that colours on adjacent tile edges match? That is, is it possible to place a tile (of one of these types) on each point of $\mathbb{N} \times \mathbb{N}$, in such a way that colours match both vertically and horizontally? For some tile types, this is possible, for others it is impossible. However there is no algorithm for deciding for which tile types this can be done; it is a simple, and elegant, example of a computationally undecidable problem. Showing that a modal logic is strong enough to encode this problem is often a straightforward way of showing its undecidability; see Blackburn, de Rijke and Venema [13] for examples of how to use the tiling problem in this way.

In a slogan: trees tend to be safe, but beware of grids. Somewhat poetically, we can imagine modal logic as a small boat navigating somewhere on the border between decidability and undecidability, as Figure 18 shows.

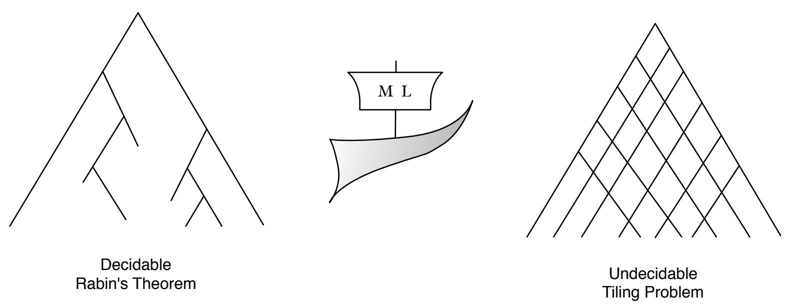

Figure 18. Modal logic: tacking between safety and danger.

Furthermore, it is important to realise that undecidable problems arise even when attention is restricted to finite frames; see, for example, Urquhart [127]. And indeed, even in the finite case, undecidability turns out to be the norm. It is straightforward to show that there are non-denumerably many distinct frame satisfiability problems over finite frame classes (an elegant demonstration of this, due to Spaan [118], is given as Exercise 6.2.4 of Blackburn, de Rijke and Venema [13]). As there are only denumerably many computable functions, undecidability is almost always guaranteed.

So what about recursive enumerability? That is, if we restrict attention to a class of frames F that is defined by a modal formula, is the theory of this frame class (that is, the set of formulas φ valid on all frames F) recursively enumerable? Well, if F is elementary, the answer is yes:

PROPOSITION 34. *Suppose that F is an elementary class of frames defined by a basic modal formula φ. Then the set of basic modal formulas that are valid on all frames in F is recursively enumerable.*

Proof. As F is an elementary class that φ defines, φ corresponds to some first-order formula α. Now a basic modal formula ψ is valid on frames for φ iff its second-order translation

$\forall P_1 \cdots P_n \forall x \mathrm{ST}_x(\psi)$ is true in all models of the first-order formula α, that is, iff

$$\alpha \models \forall P_1 \cdots P_n \forall x \mathrm{ST}_x(\psi),$$

where $\models$ is classical entailment. But as α is first-order, referring to R only, the predicates $P_1 \cdots P_n$ do not occur in α and hence this is equivalent to

$$\alpha \models \forall x \mathrm{ST}_x(\psi).$$

But this is a first-order entailment, and as such entailments are recursively enumerable the result follows. ■

However once we move beyond the elementary frame classes, even recursive enumerability is lost. A key result here is Thomason's [126] reduction of the standard consequence relation for the second-order correspondence language to the *global frame consequence* relation for a basic modal language with one modality. A basic modal formula φ is a global frame consequence of Γ if for all frames $\mathfrak{F}$, if $\mathfrak{F} \models \Gamma$, then $\mathfrak{F} \models \varphi$. It follows that global frame consequence is not recursively enumerable. Indeed, it is even Σ^1_1-complete, which means it is as hard to decide as the existential second-order theory of the natural numbers under the less-than-or-equal ordering. To put it another way: this is an example of a *highly undecidable* problem. For further discussion of Thomasons's work in this area, see Chapter 7 of this handbook.

6 RICHER LANGUAGES

So far we've been dealing almost exclusively with the basic modal language. We've seen that the key to its expressive power lies in the notion of bisimulation and that (at least when interpreted over the class of all models) it has better computational properties than first-order logic. All in all, the basic modal language is really rather elegant, so we might be tempted to ask: is it possible to lift (at least some of) its attractive properties to stronger languages? That is, can we design richer modal languages that retain, or even enhance, those features that make the basic modal language special? In fact, modal logicians have been experimenting with richer languages for years, and in this section we survey some of their work. As we shall see, this line of work adds a new dimension to our understanding of modal logic and relational semantics.

But what should count as a richer modal language? It's easier to explain what shouldn't. Here's an obvious example. It is straightforward to extend our basic definitions to cover *polyadic modalities* (that is, n-place diamonds and boxes). Simply work with models in which there is an $n+1$-place relation R^m for every n-place diamond $\langle m \rangle$. We interpret $\langle m \rangle$ using the following satisfaction clause:

$$\mathfrak{M}, w \models \langle m \rangle(\varphi_1, \ldots, \varphi_n) \quad \text{iff} \quad \begin{array}{l} \text{for some } v_1, \ldots, v_n \in W \text{ such that } R^m w v_i \ldots v_n \\ \text{we have } \mathfrak{M}, v_1 \models \varphi_1 \text{ and } \ldots \text{ and } \mathfrak{M}, v_n \models \varphi_n. \end{array}$$

Now, such n-place modalities are undeniably useful for certain purposes, especially when interpreted over restricted classes of frames. For example, when working with spatio-temporal structures, we might want to add a three place modality to capture the notion of "between", or we might want to explore the logical theory of function composition, as is done in the branch of modal logic known as arrow logic (see Marx and Venema [94]). Nonetheless, when working with the class of all models, developing the basic semantic theory (standard translation, bisimulation, and so on) of polyadic modal operators is essentially a matter of sprinkling our earlier work with additional indices.

As we shall see, the richer languages explored in this section offer much more. Moreover, their richness takes us in many different directions. Sometimes the enrichment consists of taking a standard language and insisting that a modality be interpreted by some mathematically fundamental relation (the universal modality is a good example). Sometimes the enrichment takes the form of more complex satisfaction definitions (both temporal logic with Until and Since and conditional logic are examples of this). In other cases, syntactic enhancements are introduced to support novel semantic capabilities (hybrid logic, propositional dynamic logic, and the modal μ-calculus all do this) and in one case (the guarded fragment) we enrich by abandoning modal syntax and using first-order syntax instead. Moreover, it is also possible to enrich by combining logics. For example, we might combine two propositional modal logics to enable some application domain to be more accurately modeled, or we might combine modal logic with first-order logic, a move which takes us to the historical heartland of philosophical applications of modal logic. As we shall see, modal logicians have been extremely creative when it comes to devising richer languages.

Of course, this variety raises a question of its own: what, if anything, do all these richer languages have in common? That is, what makes them all modal? This is not an easy question to answer. Nonetheless, as we work our way through this landscape a number of themes will recur: robust decidability, the importance of bisimulations, and characterisations of fragments of first- and second-order logic. As we shall see at the end of the section, the idea of restricted quantification that underlies the guarded fragment goes a long way towards accounting for these properties, for both first- and second-order enrichments. Moreover, it is possible to draw on ideas from abstract model theory and prove Lindström-style characterisation results. In short, we will often be able to lift much of the fundamental semantic theory for basic modal logic to a whole new level, a good indication that the enrichments discussed below are, in an important sense, genuinely modal.

6.1 The universal modality

Time to feed the bears again. As we said in Section 4, some problems demand a global perspective. We sometimes want to view a modal formula as a general background constraint, something that must be satisfied at *all* points in a model. Indeed, because of the importance of background constraints, in many practical situations we are primarily interested in the local-global satisfiability problem, which we formulated as follows: given basic modal formulas φ and ψ, is there a model which locally satisfies φ and globally satisfies ψ? Now, description logic, with its two level architecture of TBoxes (which impose general constraints) and ABoxes (which give information about particular individuals), acknowledges the importance of this problem (the information in a TBox has to be globally satisfied, while the information in an ABox only has to be locally satisfied). But the ability to impose global constraints is not incorporated into description logic concept languages (which are essentially notational variants of the basic modal languages we are familiar with) and this raises an interesting question. Is it possible to internalise the notion of global satisfiability in a modal language? And if so, what happens?

Let's introduce the *universal modality* and find out. To keep things simple, suppose we are working in a language with just one modality. We shall add a second modality, and will write *E* for its diamond form, and *A* for its box form. The interpretation of *E* and *A* is fixed: in any model $\mathfrak{M} = (W, R, V)$, both modalities must be interpreted using the universal relation $W \times W$. That is, the satisfaction definition for these modalities is:

$$\begin{array}{lll}\mathfrak{M}, w \models E\varphi & \text{iff} & \text{there is a } u \in W \text{ such that } \mathfrak{M}, u \models \varphi \\ \mathfrak{M}, w \models A\varphi & \text{iff} & \text{for all } u \in W \text{ we have } \mathfrak{M}, u \models \varphi.\end{array}$$

Thus $E\varphi$ scans the entire model for a point that satisfies φ, while $A\varphi$ asserts that φ holds everywhere. We have imported the meta-theoretic notion of global truth into our modal object language, or to put it another way, we have internalised the TBox. Accordingly, we call *E* the *universal diamond*, and *A* the *universal box*. If it is irrelevant whether we mean *E* or its dual, we simply talk of the *universal modality*.

How can we be sure that adding the universal modality really increases the expressive power at our disposal? That is, are we certain that *E* and *A* are not already definable in the basic modal language? We are. One way to see this is via a bisimulation argument (see Example 2.4 in Blackburn, de Rijke and Venema [13] for such a proof). But an easy complexity-theoretic argument also establishes this. Let φ and ψ be basic modal formulas. Then the formula $A\psi$ expresses the global satisfiability problem (for the basic modal language) in our new language, and the formula $\varphi \wedge A\psi$ expresses the local-global satisfiability problem (for the basic modal language) again in our new language. Now, we remarked in Section 4 that both these problems are EXPTIME-complete. However the satisfiability problem for the basic modal language is PSPACE-complete. Hence (assuming that PSPACE is strictly contained in EXPTIME , the standard assumption) our ability to express these problems in the enriched language shows that the apparent increase in expressive power is genuine.

This in turn raises a new question. Because it can encode these problems, the satisfiability problem for the enriched language is at least EXPTIME-hard. But are some problem-instances even harder? No. Everything is solvable in EXPTIME.

THEOREM 35. *The satisfiability problem for the basic modal language enriched with the universal modality is EXPTIME-complete.*

Proof. See Hemaspaandra [65], or her earlier PhD thesis Spaan [118]. ■

But the universal modality not only gives us extra expressivity at the level of models, it also increases our ability to define new classes of frames. Moreover, an elegant variant of the Goldblatt-Thomason Theorem holds for the enriched language. We'll discuss this result shortly, but let's first consider two examples of newly definable frame classes.

The class of frames of cardinality less than or equal to some natural number n (that is, frames in which $|W| \leq n$) is not definable in the basic modal language. Why not? Because basic modal validity is closed under the formation of disjoint unions. Hence any basic modal formula φ which allegedly defined this frame class could easily be shown to fail: simply by sticking together enough frames we could validate φ on frames of cardinality greater than n.

But this condition *is* definable with the help of the universal modality:

$$\bigwedge_{i=1}^{n+1} Ep_i \to \bigvee_{i \neq j} E(p_i \wedge p_j).$$

As the reader can easily check, this formula is valid on any frame where $|W| \leq n$, and can be falsified on any larger frame (in essence, the formula encodes the pigeonhole principle for $n+1$ pigeons and n holes). It follows that validity in the enriched language is not preserved under the formation of disjoint unions. This, of course, is as it should be. We want a genuine *universal* modality, not something that can be fooled by the addition of new components.

Here's a second example. The condition $\forall x \exists y\, Ryx$ (that is, every point has a predecessor) is not definable in basic modal logic. Why not? Because modal validity is preserved under the formation of generated subframes. Any basic modal formula which putatively defined this class would have to be valid on the frame $(\mathbb{N}, R)$, where Rnm iff $n > m$, the natural numbers under the reverse ordering. But (by preservation under generated subframes) it would then have to be valid on the subframe generated by any number n. But in any such subframe, n has no predecessor, hence the condition is not basic modal definable.

But it *is* definable with the help of the universal modality:

$$p \rightarrow E\Diamond p.$$

It is easy to check that this formula defines the required condition, hence it follows that validity in the enriched language is not preserved under generated subframes. Again, this is the way it should be. A genuinely universal modality will not let us throw away points: its purpose is to keep an eye on the entire frame. It should be intolerant of both additions (disjoint unions) and deletions (generated submodels).

And now for the promised result: when it comes to defining elementary frame classes, intolerance towards disjoint unions and generated submodels is precisely what distinguishes the enriched language from the basic modal language. The following result is the Goldblatt-Thomason Theorem for the basic modal language, with closure under disjoint unions and generated subframes stripped away:

THEOREM 36. *A first-order definable class of frames is definable in the basic modal language enriched with the universal modality iff it is closed under taking bounded morphic images, and reflects ultrafilter extensions.*

Proof. See Goranko and Passy [61]. ■

Three comments. First, adding the universal modality also increases our ability to define non-elementary frame classes. For example, the class of frames where the converse of the accessibility relation R is well-founded (that is, where it is impossible to form infinite R-successorship chains) is not definable in basic modal logic. Löb's formula, $\Box(\Box p \rightarrow p) \rightarrow \Box p$ doesn't quite pin this condition down (recall that it defines the conjunction of transitivity and converse well foundedness). But the following Löb-like formula in the enriched language does:

$$A(\Box p \rightarrow p) \rightarrow p.$$

(This example is from Goranko and Passy [61], the key reference on the universal modality.) Second, it is straightforward to extend the definition of bisimulation so that it works for the basic modal language enriched with the universal modality; all that needs to be done is to insist that the bisimulation be *total*, that is, that every element in each model is related to at least one point in the other; see de Rijke [30] for a brief discussion. Third, the universal modality has a big brother, the *difference operator*. The diamond form of this operator is written D, and $D\varphi$ is satisfied at a point w in a model if and only if φ is satisfied at some *different* point v (that is, the difference operator is interpreted using the $\neq$ relation on W). The difference operator is strong enough to define the universal modality ($E\varphi$ is just $\varphi \vee D\varphi$) but D cannot be defined using E (we leave the proof as an exercise). The difference operator arises naturally in many settings and, like the universal modality, has a smooth meta-theory; see de Rijke [29] for more information.

6.2 Hybrid logic

Basic modal languages have an obvious expressive weakness: they cannot name points. We cannot say this happened *then*, or that some *particular* individual has some property, or that two distinct sequences of processes take us from the current state to the *same* state. For example, in Figure 4 we let the nodes represent particular individuals such as Terry and Judy — but the basic modal language doesn't let us pick out these individuals. First-order logic, of course, lets us do this. We use constants to name individuals of interest, and the equality symbol for reasoning about their identity. No analogous mechanisms exist in basic modal logic. The *basic hybrid language* is the result of adding them.

At the heart of hybrid logic lies a simple idea, first introduced by Arthur Prior [104, 105] in the 1960s: sort the proposition symbols, and use *formulas as terms*. Let's do this right away. Take a language of basic modal logic (with proposition symbols p, q, r, and so on) and add a second sort of proposition symbol. The new symbols are called *nominals*, and are typically written i, j, k, and l. Both types of proposition symbol can be freely combined to form more complex formulas in the usual way. And now for the key change: *insist that each nominal be true at exactly one point in any model*. That is, insist (for any valuation V and nominal i) that $V(i)$ be a singleton set. We call the unique point in $V(i)$ the *denotation* of i. A nominal 'names' its denotation by being true there and nowhere else.

This change is far from negligible: already we have a more expressive logic. Consider the following basic modal formula:

$$\Diamond(r \wedge p) \wedge \Diamond(r \wedge q) \rightarrow \Diamond(p \wedge q).$$

This formula can be falsified, as the p-witnessing and q-witnessing points given by the antecedent may be distinct. But now consider the following hybrid formula:

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q).$$

This is identical to the preceding formula, except that we have replaced the proposition symbol r by the nominal i. But the resulting formula is valid. For now we have extra information: the p-witnessing and q-witnessing successors both make i true, so they are true at the same point, namely the denotation of i.

The addition of nominals is the crucial step towards the basic hybrid language, but we need a second ingredient too: *satisfaction operators*. These are operators of the form $@_i$, where i is a nominal. The formula $@_i\varphi$ asserts that φ is satisfied at the (unique) point named by the nominal i. That is:

$$\mathfrak{M}, w \models @_i\varphi \quad \text{iff} \quad \mathfrak{M}, u \models \varphi, \text{ where } u \text{ is the denotation of } i.$$

Syntactically, satisfaction operators are modalities. And they are semantically well behaved. For a start, all instances of the modal distribution schema are valid:

$$@_i(\varphi \rightarrow \psi) \rightarrow (@_i\varphi \rightarrow @_i\psi).$$

Moreover, satisfaction operators also admit the modal generalisation law: if φ is valid, then so is $@_i\varphi$ (for any choice of i). Hence satisfaction operators are normal modal operators. Moreover, they are self-dual modalities, for all instances of $@_i\varphi \leftrightarrow \neg @_i \neg\varphi$ are valid. So we are free to regard satisfaction operators as either boxes or diamonds.

But for present purposes, the most important point about satisfaction operators is that they give us a modal perspective on the equality relation. To see this, note that formulas like $@_i j$ are well formed. What does this formula assert? It says that "at the denotation of i, the nominal j is satisfied", or to put it another way, "the point named i is identical to the point named j". Hence the following schemas are valid: $@_i i$ (reflexivity of equality), $@_i j \to @_j i$ (symmetry of equality), $@_i j \wedge @_j k \to @_i k$ (transitivity of equality), and $@_i \varphi \wedge @_i j \to @_j \varphi$ (replacement). As we hoped, a modal theory of equality is emerging.

We will shortly characterise this theory, but before doing so let's glance at what is happening at the level of frames. Here too there is an increase in expressivity. None of the four first-order definable frame conditions listed below can be defined in basic modal logic. But it is easy to check that each is defined by the hybrid formula written next to them:

$\forall x \neg Rxx$	$i \to \neg\Diamond i$	(irreflexivity)
$\forall xy(Rxy \to \neg Ryx)$	$i \to \neg\Diamond\Diamond i$	(asymmetry)
$\forall xy(Rxy \wedge Ryx \to x = y)$	$i \to \Box(\Diamond i \to i)$	(antisymmetry)
$\forall xy(Rxy \vee x = y \vee Ryx)$	$@_j \Diamond i \vee @_j i \vee @_i \Diamond j$	(trichotomy).

And now for the main result. Hybridisation has given us some sort of modal theory of equality. But how much of the corresponding first-order theory have we captured? Of course, now when we talk about "corresponding first-order theory" we mean: theory in the first-order correspondence language *enriched with constants and the equality symbol*.

The first step towards an answer is to extend the standard translation to cover nominals and satisfaction operators. So enrich the first-order correspondence language with constants and the equality symbol; to keep the notation uncluttered, we'll re-use the nominals as first-order constants. Then add the following clauses to the standard translation:

$$\begin{array}{rcl} \mathrm{ST}_x(i) & = & (x = i) \\ \mathrm{ST}_x(@_i \varphi) & = & \mathrm{ST}_i(\varphi). \end{array}$$

That is, nominals i are translated into first-order constants i, and satisfaction operators are translated by substituting the relevant first-order constant for the free-variable x. Note that this translation returns first-order formulas with *at most* one free variable x, not exactly one. This is because a constant may be substituted for the free occurrence of x. For example, the hybrid formula $@_i i$ translates into the first-order *sentence* $i = i$.

The second step is to extend the notion of bisimulation given in Definition 5 to make it suitable for the basic hybrid language and for the constant-enriched first-order correspondence language:

DEFINITION 37 (Bisimulation-with-names). A bisimulation-with-names between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols, and the same nominals.

Zig: if Rwv, then there exists a point v' (in $\mathfrak{M}'$) such that vEv' and $R'w'v'$, and

Zag: if $R'w'v'$, then there exists a point v (in $\mathfrak{M}$) such that vEv' and Rwv.

Closure: All points named by nominals are related by E.

It is easy to check that all basic hybrid formulas are invariant under bisimulations-with-names; the proof is an easy extension of Lemma 9. More interestingly, such bisimulations also give rise to a Characterisation Theorem:

THEOREM 38 (Hybrid Characterisation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$ in at most one free variable x:*

1. $\varphi(x)$ *is invariant for bisimulation-with-names.*

2. $\varphi(x)$ *is equivalent to the standard translation of a basic hybrid formula.*

Proof. That clause 2 implies 1 is a more or less immediate. The hard direction is showing that clause 1 implies 2. The original proof can be found in Areces, Blackburn and Marx [6]. ■

In short, basic hybrid logic is a simple notation for capturing *exactly* the bisimulation-invariant fragment of first-order logic with constants and equality, or to put it another way, basic hybridisation is a mechanism for equality reasoning in propositional modal logic. And it comes cheap. Up to a polynomial, the complexity of the resulting decision problem is no worse than for the basic modal language we started with:

THEOREM 39. *The satisfiability problem for the basic hybrid language over arbitrary models is PSPACE-complete.*

Proof. See Areces, Blackburn and Marx [6]. ■

A number of stronger hybrid languages have also been explored. One of the most interesting extensions is to add $\downarrow$ (the *downarrow binder*). This binds occurrences of nominals within its scope to the point of evaluation. That is, to evaluate $\mathfrak{M}, w \models \downarrow i.\varphi$, we evaluate $\mathfrak{M}, w \models \varphi$ but with all occurrences of the nominal i that were bound by $\downarrow$ now interpreted as naming w (for details on how to make this informal explanation precise, see Chapter 14 of this Handbook). To put it another way, $\downarrow$ lets us create a name for *here*, and this immediately increases the expressive power at our disposal. For example, in any model $\mathfrak{M}$, the formula $\downarrow i.\neg\Diamond i$ is true at precisely the irreflexive points; as we noted earlier, no such formula exists in the basic modal language, and indeed, no such formula exists in the basic hybrid language either.

Moreover, $\downarrow$ interacts beautifully with $@$. Intuitively, $\downarrow$ stores new values for nominals, and @ allows us to retrieve them. As an example of this interaction, consider the following formula which is true in any model at points with at least two successors:

$$\downarrow i.\Diamond\downarrow j.@i\Diamond\neg j.$$

This formula first names the point of evaluation i, it then declares that i has a successor which it names j, and then (with the help of @) it jumps back to i to assert that i also has a successor distinct from j.

But this increased expressivity comes at a price: by introducing $\downarrow$ we have sailed over the border into undecidability. As we remarked earlier, the ability to create grid-like models is a useful warning sign of undecidability, and the smooth interaction between $\downarrow$ and @ makes it easy to create the unit squares required to build grids:

$$\downarrow i.\Diamond(\neg i \wedge \downarrow j.\Diamond(\neg i \wedge \neg j \wedge \downarrow k.@i\Diamond(\neg i \wedge \neg j \wedge \neg k \wedge \downarrow l.\Diamond k))).$$

If you work through this formula you will see that it demands the existence of four distinct points, which it calls i, j, k, and l, such that Rij, Rjk, Ril and Rlk. Note the characteristic use of the embedded $@_i$ to jump us back to the original point of evaluation i; this enables us to construct a second path from i to k that goes via point l. Of course, moving from this observation to a proof that it is possible to code the tiling problem takes more work, but it can be done, and the upshot is: adding $\downarrow$ has moved us up to an undecidable fragment of first-order logic.

But which fragment? The answer has two natural formulations. The first has the now-familiar form of a Characterisation Theorem: it turns out that adding downarrow has moved us up to precisely that fragment of first-order logic which is *invariant under generated submodels*. The second answer has a more syntactic flavour: we have moved up to the *bounded fragment* of first-order logic. The bounded fragment consist of all first-order formulas built up from atomic formulas using the booleans and bounded quantifications of the form $\exists y(R\tau y \wedge \varphi)$ and $\forall y(R\tau y \rightarrow \varphi)$, where τ is a term that does not contain y. The bounded fragment arises naturally in set theory (see Levy [89]) and arithmetic (see Buss [20]). In the mid-1960s, Feferman and Kreisel [41, 40] characterised the bounded fragment as the fragment of first-order logic invariant under generated submodels. It is intriguing that hybrid logic should have arrived at the same fragment by such a different route.

For full formulations and proofs of these results, see Areces, Blackburn and Marx [6]. For a detailed overview of hybrid logic, covering the results mentioned and much else besides, see Chapter 14 of this handbook.

6.3 Temporal logic with Until and Since operators

We turn now to another historically early enrichment: the addition of the binary U (Until) and S (Since) operators. These were introduced in the late 1960s by Hans Kamp [76], who added them to Arthur Prior's basic (F and P based) tense logic, and proved an elegant result: U and S are expressively complete with respect to Dedekind complete strict total orders (we discuss Kamp's result below). But, beautiful though this is, it is not what led to the present popularity of these operators. Rather, around 1980, Gabbay, Pnueli, Shelah and Stavi [53] observed that Until offers precisely what is required to state what computer scientists call *guarantee properties*, and this led to its widespread adoption for reasoning about programs. Given the number of researchers currently active in temporal logic for program verification, Until may well be the best known and most widely used modal operator of all: it plays a key role in LTL (Linear Time Temporal Logic), CTL (Computational Tree Logic), and CTL* (a highly expressive system that contains both LTL and CTL as sublogics). For an introduction to these logics, see Chapter 11 of this handbook, or Clarke, Grumberg and Peled [25].

Now, we briefly met the Until operator in Section 4 when we discussed model checking. There we defined it in terms of R^+ and R^*, the transitive and reflexive transitive closures of the underlying relation R used by the $\Diamond$ over tree-like models. Here we shall define Until and Since in their most general form:

$\mathfrak{M}, w \models U(\varphi, \psi)$	iff	there is a v such that Rwv and $\mathfrak{M}, v \models \varphi$, and for all u such that Rwu and Ruv we have $\mathfrak{M}, u \models \psi$.
$\mathfrak{M}, w \models S(\varphi, \psi)$	iff	there is a v such that Rvw and $\mathfrak{M}, v \models \varphi$, and for all u such that Rvu and Ruw we have $\mathfrak{M}, u \models \psi$.

Putting this in words, Until asserts that there is *some* point in the future where φ holds, and that at *all* points between the point of evaluation and this future φ-witnessing point, ψ holds. Since functions in the same way, but towards the past. Note the $\exists\forall$ pattern of quantification in the satisfaction definitions. These operators are neither diamonds nor boxes; they are something new and (as we shall see) more powerful.

What can we say with them? For a start, they have all the power of ordinary diamonds: $U(\varphi, \top)$ has the same meaning as $\Diamond\varphi$. But now we can say more: these operators are tailor-made for stating guarantee properties, requirements of the form "*Some event will happen, and* *until that event takes place, a certain condition will hold*". For if we represent the event by φ and the condition by ψ, then $U(\varphi, \psi)$ clearly captures what is required.

But how can we be sure that we can't state guarantee requirements in the basic modal language? A simple bisimulation argument demonstrates this. Consider the two models shown in Figure 19. The two models are clearly bisimilar (simply link both points in the right-hand model

Figure 19. Until is not definable in basic modal logic.

to the single point in the left-hand model; all proposition symbols are false at all points in both models, though this is irrelevant to the following argument). This means that the two models agree on the truth of all basic modal formulas at all points. But the models disagree on the value of $U(\top, \bot)$. This formula is false in the model on the left, but true at both points in the model of the right. We conclude that no basic modal formula can capture the effect of Until.

But this is a little too easy. Until is typically used for temporal reasoning tasks, and the two models just shown have little to recommend them as flows of time. But it turns out that Until cannot be defined even if we work with models with more structure. For a start, even if we restrict our attention to transitive models, Until is not basic modal definable. For consider the two models shown in Figure 20; we are interested in the transitive closure of the relation indicated by the arrows. These models are bisimilar (link w_0 and w_1 with w', link t_0 and t_1 with t', and so on). So suppose that there is some formula in the basic modal language that captures the effect of $U(p, q)$. Any such formula would be true in the left-hand model at points w_0 and w_1. For consider what happens at w_0 (the argument for w_1 is analogous). There is a point to its future (namely v_1) that makes p true and at all points lying in between (and there is only one, namely u) we have that q is satisfied. However any such formula would be *false* in the right-hand model at w', for here there are *two* points between w' and v' (namely u' and t') and t' does not satisfy q. As w' is bisimilar to w_0 and w_1, we conclude that no basic modal formula can capture the effect of Until. And this result can be strengthened. Even if we restrict ourselves to linear models, the basic modal language can't define Until, and it can't do so on the real numbers either (see Proposition 7.10 in Blackburn, de Rijke and Venema [13]).

So adding S and U to the basic modal language yields new expressivity — but how much? We shall now discuss Kamp's Theorem, which shows that on certain classes of structures (a class that includes the real numbers) these operators capture the entire one free variable fragment of the first-order correspondence language. This result was one of the earliest (and is still one of the most striking) purely semantic results in modal logic.

First, note that Until and Since correspond to fragments of the familiar first-order correspondence language that we have been working with throughout the chapter. After all, we can translate them as follows:

$$\begin{aligned} \mathrm{ST}_x(U(\varphi,\psi)) &= \exists z\,(Rxz \wedge \mathrm{ST}_z(\varphi) \wedge \forall y\,(Rxy \wedge Ryz \rightarrow \mathrm{ST}_y(\psi))) \\ \mathrm{ST}_x(S(\varphi,\psi)) &= \exists z\,(Rzx \wedge \mathrm{ST}_z(\varphi) \wedge \forall y\,(Rzy \wedge Ryx \rightarrow \mathrm{ST}_y(\psi))). \end{aligned}$$

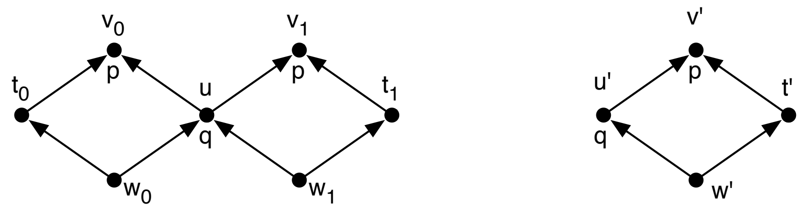

Figure 20. Even on transitive frames, Until is not definable in basic modal logic.

Incidentally, observe that we need three variables to specify this translation, whereas we only needed two for the basic modal language. Now, the three variable fragment of first-order logic is known to be undecidable, thus the translation doesn't give us an easy decidability result for the enriched modal language, though its satisfiability problem over arbitrary models is in fact decidable. We'll understand why a little later when we discuss the packed fragment.

So what does Kamp's Theorem say? First some preliminary definitions. Let M be a class of models. We say that a modal language is *expressively complete over* M, if every formula (in one free variable) from the first-order correspondence language is equivalent to a formula in the modal language (when we restrict attention to models from M). Which class of models is Kamp's Theorem about? A *strict total order* is any frame (with one binary relation R) that is transitive, irreflexive, and linear (that is, $\forall xy(Rxy \vee x = y \vee Ryx)$). A strict total order is *Dedekind complete* if every subset with an upper bound has a least upper bound. Standard examples of Dedekind complete strict total orders are the real numbers $(\mathbb{R}, <)$ and the natural numbers $(\mathbb{N}, <)$ under their usual orderings. And now we have:

THEOREM 40 (Kamp's Theorem). *The basic modal language enriched with U and S is expressively complete with respect to models based on Dedekind complete strict total orders.*

Proof. The original proof is in Kamp's thesis [76]. Elegant modern proofs, and proofs of stronger expressive completeness results, can be found in Gabbay, Hodkinson and Reynolds [52]. See also Chapter 11 of this handbook. ■

Much more could be said about the Until and Since operators, but we will confine ourselves to the following remark. Because of their $\exists\forall$ pattern of quantification, for some time it was unclear how best to define a suitable notion of bisimulation. However Kurtonina and de Rijke [87] and Sturm [120] have given definitions which enable characterisation theorems to be proved.

6.4 Conditional logic

Although formulas of the form $\varphi \to \psi$ are often glossed as "if φ then ψ", the truth conditions that classical logic gives to the $\to$ symbol (and in particular, the fact that $\varphi \to \psi$ is true when φ is false) means that $\to$ does not mirror the more interesting meanings that conditionals can have in natural language. This has inspired numerous attempts to introduce conditional connectives (say, $>$) that better mimic the logic(s) of natural language conditionals. Indeed, such aspirations have given birth to an entire branch of logic, namely Relevance Logic, which nowadays is a well-established branch of the study of substructural logics (see Restall [108]).

But there is a modal approach to conditionals too. Its motivation comes from the following intuition: a conditional $\varphi > \psi$ can (often) be read as an *invitation* to assume the antecedent (perhaps making some adjustments to accommodate its truth) and check if the consequent is true. A characteristic inferential feature of this reading is the failure of *monotonicity in the antecedent*. "If I catch the 6.22 train at Amsterdam Central (φ), I will be home on time (ψ)" is true on the most natural reading of the conditional, but adding an unusual further condition may make it false, as the sentence "If I catch the 6.22 train at Amsterdam Central (φ), and the dikes break (θ), I will be home on time (ψ)" demonstrates.

Models for modal-style conditional reasoning are triples $\mathfrak{M} = (W, C, V)$. Here W is a non-empty set (whose elements are usually called worlds), V is a valuation, and C is a ternary relation of *relative similarity* , or (as it is sometimes put in the literature) a relation of relative 'comparison' or 'preference' between worlds. It is useful to write $Cwuv$ as $C_w uv$ and to read this as saying that "w has more in common with u than v". It is standard to demand that each C_w satisfies $\forall uvz(C_w uv \wedge C_w vz \rightarrow C_w uz)$, w-centred transitivity, and $\forall u C_w uu$, w-centred reflexivity. Moreover, some authors, most famously David Lewis, also demand w-centred comparability, that is, $\forall uv(C_w uv \vee C_w vu)$. A good way to visualise the relation $C_w uv$ is to think of u and v as two concentric circles around w. If u and v are distinct, then u is a concentric circle *closer* to w than v is.

The simplest truth condition for conditionals is the following, which come from David Lewis's groundbreaking book "Counterfactuals" [90]. It fits in well with our intuitions (at least on finite models):

$\mathfrak{M}, w \models \varphi > \psi$ iff all minimal worlds in the w-centred ordering $C_w uv$ at which φ is true are also worlds where ψ holds.

This satisfaction clause can be phrased more succinctly as follows: all minimal φ-worlds are ψ-worlds.

Note that the φ*-minimal* worlds around w are the only ones we consider. As the minimal worlds satisfying the stronger condition $\varphi \wedge \theta$ need not be the ones satisfying φ, in this way we get a semantic distinction which accounts for the failure of monotonicity in the antecedent.

But what about *infinite* models? Then there need not be any minimal worlds satisfying the antecedent (we might have a chain of φ-satisfying concentric circles coming ever closer to w). Here's a way of handling this: switch to the following more complex truth condition (to keep things readable, we shall write use $\varphi(v)$ as shorthand for $\mathfrak{M}, v \models \varphi$, and similarly for ψ):

$$\mathfrak{M}, w \models \varphi > \psi \quad \text{iff} \quad \forall u(\varphi(u) \Rightarrow \exists v(C_w vu \;\&\; \varphi(v) \;\&\; \forall z((C_w zv \;\&\; \varphi(z)) \Rightarrow \psi(z)).$$

This says that the conditional $\varphi > \psi$ holds if, whenever φ holds at some circle u, then there is some smaller circle v where φ holds such that all circles z within v satisfy ψ. This is rather awkward to process in first-order logic, but it can be clearly expressed in modal logic if we make use of a unary modality $\langle c \rangle$ (which looks inwards for a circle closer to the centre) together with the universal modality A. For then we can simply say:

$$\varphi > \psi \;=_{def}\; A(\varphi \rightarrow \langle c \rangle(\varphi \wedge [c](\varphi \rightarrow \psi)).$$

This more complex truth-condition validates a minimal logic which includes such principles as upward monotonicity in the consequent: $\varphi > \psi$ implies $\varphi > (\psi \vee \theta)$. Further properties of the similarity ordering enforce special axioms via standard frame correspondences. Assuming just reflexivity and transitivity yields the minimal conditional logic originally axiomatised by Burgess [19] and Veltman [143], while assuming also comparability of the ordering gives rise to the logics obtained by Davis Lewis.

What about complexity? A number of interesting results are known:

THEOREM 41. *The satisfiability problem for the minimal conditional logic (that is, where $C_w uv$ is transitive and reflexive) is PSPACE-complete when formulas with arbitrary nestings of conditionals are allowed, and NP-complete for formulas with bounded nesting of conditionals.*

Proof. See Friedman and Halpern [50]. These authors also prove that if uniformity is assumed (that is, if all worlds agree on what worlds are possible) the complexity rises to EXPTIME-complete, even for formulas with bounded nesting. Moreover, they show that if absoluteness is assumed (that is, all worlds agree on all conditional statements) the decision problem is NP-complete for formulas with arbitrary nesting. ■

In general, conditional logic has not been studied semantically in the same style as most modal languages, though there is no reason why it cannot be. For example, bisimulations could be defined for $>$ in much the same spirit as they are defined for temporal logics with Until and Since. Likewise, issues of frame definability beyond the minimal setting can be explored; for example, van Benthem [137] notes correspondences between conditional axioms and triangle inequalities concerning concrete geometrical relations of relative nearness in space. Many recent technical developments in conditional logic, however, have to do with its connection with *belief revision theory* (see Gärdenfors and Rott [55]). In that setting, a conditional $\varphi > \psi$ means "if I revise my current beliefs with the information that φ, then ψ will be among my new beliefs"; see, for example, Ryan and Schobbens [110]. For more on these topics, see Chapters 20 and 21 of this handbook.

6.5 The guarded fragment

The richer modal languages so far examined have clearly been modal in a syntactic sense; all use the typical "apply operator to formula" syntax. The guarded fragment, however, arises as an attempt to *directly* isolate fragments of first-order logic that can plausibly be called modal. So the modal languages we shall consider here are syntactically first-order.

The clue leading to the guarded fragment is the standard translation of the modalities. This treats modalities as macros embodying *restricted* forms of first-order quantification, in particular, quantification restricted to successor states:

$$\begin{array}{rcl} \mathrm{ST}_x(\Diamond\varphi) & = & \exists y(Rxy \wedge \mathrm{ST}_y(\varphi)) \\ \mathrm{ST}_x(\Box\varphi) & = & \forall y(Rxy \rightarrow \mathrm{ST}_y(\varphi)). \end{array}$$

As we saw earlier, it is this restricted form of quantification that lets bisimulation emerge as the key model-theoretic notion. And bisimulation, via the tree model property, leads to decidability. Thus at least one pleasant property of modal logic can plausibly be traced back to its use of a restricted form of quantification. So it is natural to ask whether other first-order fragments defined by restricted quantification have such properties. This line of enquiry leads to the guarded fragment and its relatives.

The first step takes us to the guarded fragment, which was introduced by Andréka, van Benthem, and Németi [5]. Guarded formulas φ are built up as follows:

$$\varphi \quad ::= \quad Q\overline{x} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \exists\overline{y}(G(\overline{x},\overline{y}) \wedge \varphi(\overline{x},\overline{y})) \mid \forall\overline{y}(G(\overline{x},\overline{y}) \rightarrow \varphi(\overline{x},\overline{y})).$$

Here $\overline{x}$ and $\overline{y}$ are finite tuples of variables, Q is a predicate symbol (of appropriate arity for the tuple $\overline{x}$), and G, the symbol used in the guard, is a predicate symbol too (thus the guard is an atomic formula). The key point to observe is that in the clauses for the quantifiers, all the free variables of φ appear in the guard. The set of all guarded first-order formulas is called the guarded fragment.

THEOREM 42. *The guarded fragment is decidable. Its satisfiability problem is 2EXPTIME-complete, and EXPTIME-complete if we have a fixed upper bound on the arity of predicates. Moreover, the guarded fragment has the finite model property.*

Proof. See Grädel [62] for the complexity results and a direct proof of the finite model property. An earlier (algebraic) proof of the finite model property can be found in Andréka, Hodkinson, and Németi [4]. ■

The guarded fragment is a natural generalisation of the first-order formulas obtainable under the standard translation, but does it go far enough? For example, adding Until to a basic modal language yields a decidable logic, but the standard translation of $U(p,q)$, namely

$$\exists y\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy) \rightarrow Qz)),$$

does not belong to the guarded fragment, and it can be shown that it is not equivalent to a formula in the guarded fragment either. This suggests that it may be possible to pin down richer restricted-quantification first-order fragments that retain decidability, and several closely related extensions of the guarded fragment, such as the loosely guarded fragment (see van Benthem [135]) and the packed fragment (see Marx [93]) have been proposed which do precisely this. Let's take a quick look at the packed fragment.

The packed fragment allows us to use *composite guards* γ instead of just atomic guards G. Let γ be a formula whose free variables are $\{x_1,\ldots,x_k\}$. Then γ *packs* $\{x_1,\ldots,x_k\}$ if γ is a conjunction of formulas of the form $x_i = x_j$, $R(x_{i_1},\ldots,x_{i_n})$ or $\exists\overline{x}R(x_{i_1},\cdots,x_{i_n})$, and moreover, for any two distinct free variables x_i and x_j, there is a conjunct in γ in which they both occur free. The packed fragment is the smallest fragment of modal logic that contains all atomic formulas, and is closed under boolean combinations and *packed quantification*. That is, if ψ is a packed formula, and γ packs ψ, and all the free variables of ψ are free in γ, then $\exists\overline{x}(\gamma \wedge \phi)$ and $\forall\overline{x}(\gamma \rightarrow \phi)$ are packed too.

As an example, consider again the standard translation of $U(p,q)$, namely

$$\exists y\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy) \rightarrow Qz)).$$

This is not packed as the guard of the subformula $\forall z\,((Rxz \wedge Rzy) \rightarrow Qz))$ has no conjunct in which x and y occur together. But this is easy to fix. The following (logically equivalent) formula *is* packed:

$$\exists x\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy \wedge Rxy) \rightarrow Qz)).$$

And indeed, the packed fragment turns out to be computationally well behaved:

THEOREM 43. *The packed fragment is decidable. Its satisfiability problem is 2EXPTIME-complete. Moreover, it has the finite model property.*

Proof. The complexity result follows from results in Grädel [62]. The original proof of the finite model property for the packed fragment (and the loosely guarded fragment) can be found in Hodkinson [68]; a more elegant proof can be found in Hodkinson and Otto [69]. ■

In short, we have isolated two decidable fragments of first-order logic which are expressive enough to generalise many common modal languages. Moreover, these fragments have attractive properties besides decidability. Basic modal logic resembles first-order logic in most of its meta-properties, even those (such as Craig Interpolation, Beth definability, and the standard model-theoretic preservation theorems) that do not follow straightforwardly from the fact that it is a first-order fragment. The guarded fragment shares this good behaviour to some extent, witness the Łos-style preservation theorem for submodels given in Andréka, van Benthem, and Németi [5]. But subsequent work has shown that the picture is somewhat mixed. There is indeed a natural notion of guarded bisimulation (see Andréka, van Benthem, and Németi [5]) which characterises the guarded fragment as a fragment of first-order logic. Moreover, Beth definability holds (see Hoogland, Marx and Otto [71]). However Craig interpolation fails in its strong form, though it holds when we view guard predicates as part of the logical vocabulary (see Hoogland and Marx [70]).

This is a good moment to take stock of some of the first-order fragments we have encountered in the course of this chapter, and their interrelationships. Figure 21 summarises the relationships

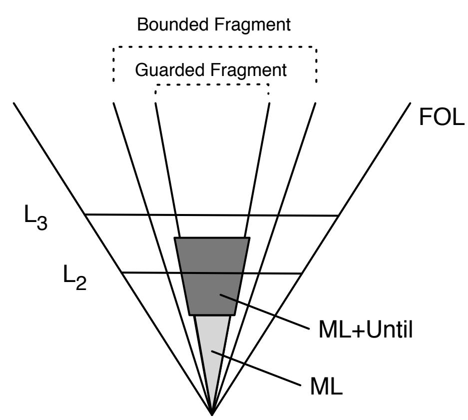

Figure 21. Some modally significant fragments of first-order logic.

between first-order logic, the more restricted (but undecidable) bounded fragment, and the still more restricted (but decidable) guarded fragment. Also shown are the fragments of first-order logic corresponding to the basic modal language, and the fragment corresponding to the basic language enriched with Until. Here L_2 and L_3 indicate the two and three variable fragments respectively; the basic language fits into the former, but the Until enriched language spills over into the latter.

6.6 Propositional Dynamic Logic

The richer modal languages so far discussed extend the first-order expressive power available for talking about models: the universal modality adds quantification over $W \times W$, hybridisation gives access to constants and equality, Until and Since and conditional logic add richer quantificational patterns, and the guarded-fragment cheerfully replaces modal syntax with first-order syntax. But the next two languages we shall discuss take us in a different direction: both add *second-order* expressive power. Now, in Section 5 we saw that modal languages have second-order expressive power (via the concept of validity) at the level of *frames*. But in the languages we now consider, second-order expressivity arises directly: it is hardwired into the satisfaction definitions, and hence is available at the level of *models*. In particular, Propositional Dynamic Logic (henceforth PDL) offers us an (infinite collection of) transitive closure operators, and the modal μ-calculus offers us a general mechanism for forming fixed-points. Significantly, both PDL and the modal μ-calculus were born in theoretical computer science. Finite structures are crucial to the theory and practice of computation, and basic results of finite model theory (see Ebbinghaus and Flum [35]) show that first-order logic is badly behaved when interpreted over finite structures. Nowadays it is standard practice to extend first-order languages with second-order constructs (such as the ability to take transitive closures or form fixed-points) when working with finite models, and in the languages we now consider, such ideas are put to work in modal logic.

Let's start by looking at the weaker of the two languages, namely PDL. The underlying idea (to extend modal logic with a modality for every program) is due to Vaughan Pratt [102], and the language now called PDL was first investigated by Fisher and Ladner [47, 48]. PDL contains an infinite collection of diamonds. Each has the form $\langle\pi\rangle$, where π denotes a non-deterministic program. The intended interpretation of $\langle\pi\rangle\varphi$ is that "some terminating execution of π from the current state leads to a state with the information φ". The dual assertion $[\pi]\varphi$ states that "every terminating execution of π from the current state leads to a state with the information φ". Crucially, the inductive structure of programs is made explicit in PDL's syntax, as complex programs are built out of basic programs using four program constructors. Suppose we have fixed a set of basic programs a, b, c, and so on. We are allowed to define complex programs π over this base as follows:

Choice: if π_1 and π_2 are programs, then so is $\pi_1 \cup \pi_2$. It non-deterministically executes either π_1 or π_2.

Composition: if π_1 and π_2 are programs, then so is $\pi_1 ; \pi_2$. It first executes π_1 and then executes π_2.

Iteration: If π is a program, then so is π^*. It executes π a finite (possibly zero) number of times.

Test: if φ is a formula, then $\varphi?$ is a program. It tests whether φ holds, and if so, continues; if not, it fails.

Hence PDL makes available the following (inductively defined) algebra of diamonds. First we have diamonds $\langle a\rangle$, $\langle b\rangle$, $\langle c\rangle$, and so on, for working with the basic programs. Then, if $\langle\pi_1\rangle$ and $\langle\pi_2\rangle$ are diamonds and φ is a formula, $\langle\pi_1 \cup \pi_2\rangle$, $\langle\pi_1 ; \pi_2\rangle$, $\langle\pi_1^*\rangle$ and $\langle\varphi?\rangle$ are diamonds too. Note the unusual syntax of the test constructor diamond: it makes a modality out of a formula. This means that the sets of PDL formulas and modalities are defined by mutual induction.

How do we interpret PDL? Syntactically we're simply dealing with a basic modal language in which the modalities are indexed by a structured set. So a model for PDL will have the form we are used to, namely

$$(W, \{R^\pi \mid \pi \text{ is a program }\}, V),$$

a suitably indexed collection of relations together with a valuation. Moreover, the usual satisfaction definition is all that is required: diamonds existentially quantify over the relevant transitions, and boxes universally quantify over them. Nonetheless, something more needs to be said. Given the intended interpretation of PDL, most of these models are uninteresting. We want models built over frames which do justice to the intended meaning of our program constructors. Which models are these?

Nothing much needs to be said about the interpretation of the basic programs: any binary relation can be regarded as a transition relation for a non-deterministic program (though if we were interested in *deterministic* programs, we would insist on working with frames in which each basic program was interpreted by a partial function). Nor need much be said about the test operator. Unusual though its syntax is, its intended interpretation in any model $\mathfrak{M}$ is simply

$$R^{\varphi?} \quad = \quad \{(w,v) \mid w = v \text{ and } \mathfrak{M}, w \models \varphi\}.$$

But the three remaining constructors demand that we impose inductive structure on our frames. Here's what is required:

$$\begin{array}{rcl} R^{\pi_1 \cup \pi_2} & = & R^{\pi_1} \cup R^{\pi_2}, \\ R^{\pi_1 ; \pi_2} & = & R^{\pi_1} \circ R^{\pi_2} \ (= \{(x,y) \mid \exists z\, (R^{\pi_1}xz \wedge R^{\pi_2}zy)\}), \\ R^{\pi_1^*} & = & (R^{\pi_1})^*, \text{ the reflexive transitive closure of } R^{\pi_1}. \end{array}$$

These restrictions are the natural set-theoretic ways of capturing the “either-or” nature of non-deterministic choices (for $R^{\pi_1 \cup \pi_2}$), the idea of executing two programs in a sequence (for $R^{\pi_1;\pi_2}$) and the idea of iterating the execution of a program finitely many times (for $R^{\pi_1^*}$). Accordingly, we make the following definition. Let Π be the smallest set of programs containing the basic programs and the programs constructed over them using the constructors $\cup$, ;, and *. Then a *regular frame* over Π is a frame $(W, \{R^\pi \mid \pi \in \Pi\})$ where R^a is a binary relation for each basic program a, and for all complex programs π, R^π is the binary relation constructed inductively using the above clauses. A *regular model* over Π is a model built over a regular frame (that is, regular models are regular frames together with a valuation). When working with PDL over the programs in Π, we will be interested in regular models for Π, for these are the models that capture the intended interpretation. All very simple and natural — but by insisting that $R^{\pi_1^*}$ be interpreted by the reflexive transitive closure of R^{π_1}, we have given PDL genuinely *second-order* expressive power. A straightforward application of the Compactness Theorem shows that first-order logic cannot define the transitive closures of arbitrary binary relations, so with this definition we've moved beyond the confines of first-order logic. Unsurprisingly, compactness fails in PDL. To see this, consider the following infinite set of formulas:

$$\{\langle \pi^* \rangle p, \neg p, [\pi]\neg p, [\pi][\pi]\neg p, [\pi][\pi][\pi]\neg p, \ldots\}.$$

It is clear that every finite subset of this set has a regular model: we simply make p true at a state reachable by taking $n+1$ (non-reflexive) π-steps out from the current state, where n is the maximal level of nesting of boxes. But the entire set cannot be satisfied at any state in any regular model.

So we have genuine second-order expressivity at our disposal. What can we do with it? Well, for a start, at the level of models, we can express some familiar algorithmic constructs:

$$\begin{array}{rl} (p?\,;a) \cup (\neg p?\,;b) & \quad \texttt{if } p \texttt{ then } a \texttt{ else } b. \\ a; (\neg p?; a)^*; p? & \quad \texttt{repeat } a \texttt{ until } p. \\ (p?; a)^*; \neg p? & \quad \texttt{while } p \texttt{ do } a. \end{array}$$

Note the crucial role played by * in capturing the effect of the two loop constructors.

Moreover, the second-order expressivity built in at the level of models spills over into the level of frames. Here's a nice illustration. Via the concept of validity, PDL itself is strong enough to define the class of regular frames (something which cannot be done in a first-order language). Now, it is not hard to give conditions that capture choice and composition. The formula

$$\langle\pi_1 \cup \pi_2\rangle p \leftrightarrow \langle\pi_1\rangle p \vee \langle\pi_2\rangle p$$

is valid on precisely those frames satisfying $R^{\pi_1 \cup \pi_2} = R^{\pi_1} \cup R^{\pi_2}$, and

$$\langle\pi_1 ; \pi_2\rangle p \leftrightarrow \langle\pi_1\rangle\langle\pi_2\rangle p$$

is valid on precisely those frames satisfying $R^{\pi_1;\pi_2} = R^{\pi_1} \circ R^{\pi_2}$.

But these are first-order conditions. What about iteration? We demanded that the relation R^{π^*} used for the program π^* be the reflexive transitive closure of the relation R^π used for π. This constraint cannot be expressed in first-order logic; how can we impose it via PDL validity?

As follows. First we demand that

$$\langle\pi^*\rangle\varphi \leftrightarrow \varphi \vee \langle\pi \,;\, \pi^*\rangle\varphi$$

be valid. This says that a state satisfying φ can be reached by executing π a finite number of times if and only if φ is satisfied in the current state, or we can execute π once and then find a state satisfying φ after finitely many more iterations of π. Second, we demand that

$$[\pi^*](\varphi \rightarrow [\pi]\varphi) \rightarrow (\varphi \rightarrow [\pi^*]\varphi)$$

be valid too. This is called *Segerberg's axiom*. Work through what it says: as you will see, in essence it is an induction schema. A frame validates all instances of the four schemas just introduced if and only if it is a regular frame.

Summing up, at both the level of models and frames, PDL has a great deal of expressive power. Hence the following result is all the more surprising:

THEOREM 44. *PDL has the finite model property and is decidable. Its satisfiability problem is EXPTIME-complete.*

Proof. The finite model property, decidability, and EXPTIME-hardness results for PDL were proved in Fisher and Ladner [47, 48]. The existence of an EXPTIME algorithm for PDL satisfiability was proved in Pratt [103]. ■

But we are only half-way through our story. With the modal μ-calculus we will climb even higher in second-order expressivity hierarchy, and we will do so without leaving EXPTIME.

6.7 *The modal μ-calculus*

The modal μ-calculus is the basic modal language extended with a mechanism for forming least (and greatest) fixed-points. It is highly expressive (as we shall see, it is stronger than PDL) and computationally well behaved. Moreover it has a beautiful bisimulation-based characterisation. All in all, it is one of the most significant languages on the modal landscape. It was introduced in its present form by Dexter Kozen [80].

The idea underlying the modal μ-calculus is to view modal formulas as *set-theoretic operators*, and to add mechanisms for specifying their fixed-points. Now, a set-theoretic operator on a set W is simply a function $F : 2^W \mapsto 2^W$. But how can we view modal formulas as set-theoretic operators? Consider a formula φ containing some proposition symbol (say p). In any model, φ will be satisfied at some set of points. If we systematically vary the set of points that the valuation assigns to p, the set of points where φ is satisfied will typically vary too. So we can view φ as inducing an operator over the points of some model, namely the operator that takes as argument the subset of W that is assigned to p, and returns the set of points where φ is satisfied with respect to this assignment.

Let's make this precise. We will work in a language with a collection of diamonds $\langle\pi\rangle$, so models have the form $\mathfrak{M} = (W, \{R^\pi\}_{\pi\in\text{MOD}}, V)$. For any proposition symbol p, $V(p)$ is the set of points in $\mathfrak{M}$ where p is satisfied. Let's extend V to a function that returns, for arbitrary formulas φ, the set of points in $\mathfrak{M}$ that satisfy φ (we won't invent a new name for this extended valuation, we'll simply call it V). The required definition is a simple reformulation of the satisfaction definition for the basic modal language:

$$\begin{array}{rcl} V(p) & = & V(p) \text{ for all proposition symbols } p \\ V(\neg\varphi) & = & W\backslash V(\varphi) \\ V(\varphi\wedge\psi) & = & V(\varphi)\cap V(\psi) \\ V(\langle\pi\rangle\varphi) & = & \{w \mid \text{for some } v\in W,\ R^\pi wv \text{ and } v\in V(\varphi)\}. \end{array}$$

Furthermore, for any proposition symbol p and any $U \subseteq W$ we shall write $V_{[p\leftarrow U]}$ for the (extended) valuation that differs from the (extended) valuation V, if at all, only in that it assigns U to p. That is, $V_{[p\leftarrow U]}(p) = U$, and for any $q \neq p$, $V_{[p\leftarrow U]}(q) = V(q)$. Then the operator induced by a formula φ (relative to a proposition symbol p) is the function that maps any $U \subseteq W$ to $V_{[p\leftarrow U]}(\varphi)$.

Now to bring fixed-points into the picture. A subset X of W is a fixed-point of a set-theoretic operator F on W if $F(X) = X$. This is clearly a special property: which set-theoretic operators have fixed-points, and how do we calculate them? The Knaster-Tarski Theorem (see Knaster [79] and Tarski [122]) gives important answers. Firstly, this theorem tells us that fixed-points exist when we work with *monotone* set-theoretic operators (an operator F is monotone if $X \subseteq Y$ implies that $F(X) \subseteq F(Y)$). Secondly, this theorem tells us that if F is a monotone operator on a set W, then F has a least fixed-point μF, which is equal to

$$\bigcap\{U \subseteq W \mid F(U) \subseteq U\},$$

and also a greatest fixed-point νF, which is equal to

$$\bigcup\{U \subseteq W \mid U \subseteq F(U)\}.$$

That is, both μF and νF are solutions to the equation $F(X) = X$, and furthermore, for any other solution Z, we have that $\mu F \subseteq Z \subseteq \nu F$. The least and greatest fixed-points given by the Knaster-Tarski Theorem are the fixed-points the modal μ-calculus works with.

But how can we specify these fixed-points using modal formulas? By enriching the syntax with an operator μ that binds occurrences of proposition symbols. That is, we shall write expressions like $\mu p.\varphi$, in which all free occurrence of the proposition symbol p in φ are bound by μ. The intended interpretation of $\mu p.\varphi$ is that it denotes the subset of W that is the least fixed-point of the set-theoretic operator induced by φ with respect to p. Fine — but how do we know that this fixed-point exists? If φ is arbitrary, we don't. However if all free occurrences of p in φ occur positively (that is, if they all occur under the scope of an even number of negations) then a simple inductive argument shows that the set-theoretic operator induced by φ is monotone, and hence (by the Knaster-Tarski Theorem) has least (and greatest) fixed-points. Accordingly we impose the syntactic restriction that the μ operator can only be used to bind a proposition symbol when all free occurrences of the variable occur positively. With this restriction in mind we define:

$$V(\mu p.\varphi) \ = \ \bigcap\{U \subseteq W \mid V_{[p \leftarrow U]}(\varphi) \subseteq U\}.$$

That is, the set assigned to $\mu p.\varphi$ is the least fixed-point of the operator induced by φ.

What can we say with the modal μ-calculus? Consider the expression

$$\mu p.(\varphi \vee \langle\pi\rangle p).$$

Read this as defining "the least property (subset) p such that either φ is in p or $\langle\pi\rangle p$ is in p". What is this set? A little experiment will convince you that it must be

$$\{w \in W \mid \mathfrak{M}, w \models \varphi \text{ or there is a finite } R^\pi\text{-sequence from } w \text{ to } v \text{ such that } \mathfrak{M}, v \models \varphi\}.$$

(The reader should check that this set really is the one given to us by the Knaster-Tarski Theorem.) Note that this is exactly the set of points that make the PDL formula $\langle\pi^*\rangle\varphi$ true.

How do we specify greatest fixed-points? With the help of the ν operator. This is defined as follows:

$$\nu p.\varphi \ =_{\mathit{def}} \ \neg\mu p.\neg\varphi(\neg p/p),$$

where $\varphi(\neg p/p)$ is the result of replacing occurrences of p by $\neg p$ is φ. This expression is well-formed: if φ is a formula that we could legitimately apply the μ operator to (that is, if all occurrences of p occur under the scope of an even number of negations), then so is $\neg\varphi(\neg p/p)$. The reader should check that this operator picks out the following set:

$$V(\nu p.\varphi) \ = \ \bigcup\{U \subseteq W \mid U \subseteq V_{[p \leftarrow U]}(\varphi)\}.$$

That is (in accordance with the Knaster-Tarski Theorem) it picks out the greatest fixed-point of the operator induced by φ. As a further exercise, the reader should check that

$$\nu p.(\varphi \wedge [\pi]p)$$

denotes the following set:

$$\{w \in W \mid \mathfrak{M}, w \models \varphi \text{ and at every } v \text{ reachable from } w \text{ by a finite } R^\pi\text{-sequence}, \mathfrak{M}, v \models \varphi\}.$$

Note that this is exactly the set of points w that make the PDL formula $[\pi^*]\varphi$ true.

In view of these examples, it should not come as a surprise that PDL can be translated into the modal μ-calculus. Here are the key clauses:

$$\begin{array}{rcl}
(\langle\pi_1 \cup \pi_2\rangle\varphi)^{\mathit{mu}} & = & \langle\pi_1\rangle(\varphi)^{\mathit{mu}} \vee \langle\pi_2\rangle(\varphi)^{\mathit{mu}} \\
(\langle\pi_1;\pi_2\rangle\varphi)^{\mathit{mu}} & = & \langle\pi_1\rangle\langle\pi_2\rangle(\varphi)^{\mathit{mu}} \\
(\langle\pi^*\rangle\varphi)^{\mathit{mu}} & = & \mu p.((\varphi)^{\mathit{mu}} \vee (\langle\pi\rangle p)^{\mathit{mu}}), \text{ where } p \text{ does not occur in } \varphi.
\end{array}$$

In fact the modal μ-calculus is strictly more expressive than PDL. The simplest example of a construct that PDL cannot model but that the modal μ-calculus can is the *repeat* operator. The expression *repeat(π)* is true at a state w if and only if there is an infinite sequence of R^π transitions leading from w. Proving that this is not expressible in PDL is tricky, but it can be expressed in the modal μ-calculus: the formula $\nu p.\langle\pi\rangle p$ does so. Moreover, the temporal logics standardly used in computer science, such as LTL, CTL, and CTL*, can also be embedded in the modal μ-calculus. For remarks and references on this topic, see Chapter 12 of this handbook.

All in all, the modal μ-calculus is a highly expressive language. In spite of this, it is extremely well behaved, both computationally and in other respects. For a start we have that:

THEOREM 45. *The modal μ-calculus has the finite model property and is decidable. Its satisfiability problem is EXPTIME-complete.*

Proof. The original decidability proof was given in Kozen and Parikh [81]. The finite model property was first established in Street and Emerson [119]. The complexity result is from Emerson and Jutla [36]. ∎

Furthermore, experience shows that the modal μ-calculus is also well behaved when it comes to model checking — indeed it is widely believed that its model checking task can be performed in polynomial time. However, at the time of writing, this conjecture has resisted all attempts to prove it.

Moreover, the modal μ-calculus has an elegant semantic characterisation. Suppose we add the following clause to the standard translation for basic modal logic:

$$\mathrm{ST}_x(\mu p.\varphi) = \forall P(\forall y((\mathrm{ST}_x(\varphi) \rightarrow Py) \rightarrow Py)).$$

This clearly captures the intended semantics of μ. But note that by adding this clause we are viewing the standard translation as taking us to monadic second-order logic, for here we bind the unary predicate symbol P. This language is already familiar to us: it's the frame correspondence language introduced in Section 5, but here we're using it to express a correspondence at the level of *models*. Thus (even at the level of models) the modal μ-calculus is a fragment of monadic second-order logic. But which fragment? This one:

THEOREM 46 (Modal μ-Calculus Characterisation Theorem). *The modal μ-calculus is the bisimulation invariant fragment of monadic second-order logic.*

Proof. See Janin and Walukiewicz [73]. ∎

For more on the modal μ-calculus, see Chapter 12 of this handbook. As well as giving a detailed technical overview, the chapter also gives an informal introduction to thinking in terms of fixed-points, which is often a stumbling block when the modal μ-calculus is encountered for the first time.

6.8 Combined logics

We now turn to what is (at first glance) one of the simplest methods of obtaining a richer modal language: combine two pre-existing ones. But for all its apparent simplicity, this method of enrichment swiftly leads to difficult territory.

Many applications lead naturally to the idea of combined logics. A good example is planning. Planning involves a collection of agents who must reason about what they are going to do given that they know the effects of actions, and where getting more information may be important for solving the problem at hand. Hence Robert Moore [98] proposed a combined language for this task. His language offered both epistemic and action modalities, making it possible to say things like

$K_i[a]\varphi$ "agent i knows that doing a has the effect φ"

and

$$[a]K_i\varphi \quad \text{"doing } a \text{ makes agent } i \text{ know that } \varphi\text{".}$$

Actually, Moore also considered combinations of PDL with epistemic operators, as plans are usually complex actions with program structure.

The fun starts when we ask how the two logics live together. For example, should they simply live side by side, the simple fusion of the two component logics? Or are there interactions between them? Obviously this depends on what we are modeling. For example, should $K_i[a]\varphi$ imply $[a]K_i\varphi$? In general, no. After all, I may know that after drinking I am boring, but unfortunately after drinking I no longer know that I am boring (that is, drinking is not an epistemically transparent action). Nor need the converse implication hold for actions that deliver genuinely new information. After consulting my account manager, I know I am broke, but I do not know now that after the consultation I am broke.

If our application does not require the modeling of such interactions, then we are dealing with the simplest possible combination of two decidable modal logics, and the result is again decidable. But for some applications we might want to enforce these interactions. Let R_a be the accessibility relation for action a, and let $\sim_i$ be the epistemic relation for agent i. The following frame correspondences tell us what these interactions give rise to:

$$\mathfrak{F} \models K_i[a]p \to [a]K_ip \quad \text{iff } \forall xyz((R_axy \wedge y \sim_i z) \to \exists u(x \sim_i u \wedge R_auz))$$
$$\mathfrak{F} \models [a]K_ip \to K_i[a]p \quad \text{iff } \forall xyz((x \sim_i y \wedge R_ayz) \to \exists u(R_axu \wedge u \sim_i z)).$$

The first principle says that new uncertainty links between the results of an action are inherited from existing ones; this is a version of the game-theoretic principle of *perfect recall*. The other direction is called *no learning*. These are powerful interaction principles. Indeed, they impose a grid-like interaction between the relations interpreting the modalities, hence the possibility arises of showing undecidability by encoding the tiling problem. A good source of information on this topic is Halpern and Vardi [64]. Among other things they show that the combined modal epistemic logic of agents with perfect recall, though still decidable, is highly complex, and that if a common knowledge operator (that is, using PDL notation, a box of the form $[(\sim_1 \cup \cdots \cup \sim_n)^*]$) is added, the problem becomes undecidable. This is a natural example of the bad computational behaviour that combinations of relatively simple decidable modal logics can give rise to. Moreover the air of mystery ("How can a description of well behaved agents get so complex?") quickly gets dispelled once we realise that the behaviour of special agents may have a rich mathematical structure that makes their logic tough.

In recent years there has been intensive theoretical work on combinations of modal logic. The goal has been to provide general *transfer results*: given two (or more) modal logics, and a method of combining them, when do properties such as decidability, finite model property, and finite axiomatisability transfer from the component logics to the combined logic? The simplest way of combining two modal logics is to take their *fusion*. Given two modal logics L_1 and L_2 (in languages with disjoint sets of modal operators) then their fusion $L_1 \otimes L_2$ is the smallest logic L in their joint language that contains them both. Fusions of modal logic have been investigated in detail (key papers include Kracht and Wolter [82], Fine and Schurz [46], and Wolter [144]), and have some pleasant transfer properties. For example, to axiomatise the fusion logic L, it suffices to take the axioms for each of the components (that is, no interaction axioms involving modalities from both language are required). Moreover, both the finite model property and decidability transfer from the component logics to the fusion.

But this good behaviour reflects the fact that fusion is a combination method designed to minimise the interaction between the component modalities. What of combination methods which allow strong interaction between the modalities? The best studied combination technique here is the formation of *products* of modal logics. Given two frames $\mathfrak{F}_1 = (W_1, R_1)$ and $\mathfrak{F}_2 = (W_2, R_2)$, their product $\mathfrak{F}_1 \times \mathfrak{F}_2$ is the frame $(W_1 \times W_2, R_h, R_v)$. Here R_h is the binary relation on $W_1 \times W_2$ defined by $(u_1, v_1)R_h(u_2, v_2)$ iff $u_1 R_1 u_2$ and $v_1 = v_2$; and R_v is the relation defined by $(u_1, v_1)R_v(u_2, v_2)$ iff $v_1 R_2 v_2$ and $u_1 = u_2$. The idea of taking products of modal logics is an old one (dating back to at least Segerberg [114]) and is a widely used combination method in many applications of modal logic. But the product construction creates frames which allow for very strong interactions between the modalities, and there are far fewer transfer results for this method of combination; indeed, there are many negative results showing transfer of decidability failures.

Work on combination of logics, from both applied and theoretical perspectives, is one of the liveliest areas of research in contemporary modal logic. For a detailed survey of fusions, products, and methods of combinations between these extremes, see Chapter 15 of this handbook.

6.9 First-order modal logic

We turn now to what is arguably one of the least well behaved modal languages ever proposed: first-order modal logic. However, in one of those twists that make intellectual history so fascinating, first-order modal logic has come to be accepted (at least in philosophical quarters) as the most important modal logic of all. For many philosophers, modal logic *is* first-order modal logic.

This is not to say that first-order modal logic is philosophically uncontroversial. Indeed, as is discussed in Chapter 21 of this handbook, one of the liveliest debates in 20th century analytic philosophy was ignited when Quine [106] questioned the coherence of the enterprise. But two advances led to its acceptance. The first was the development of the relational semantics of first-order modal logic (Kripke [83, 85] are key papers here) and the second was the publication of “Naming and Necessity” (Kripke [86]) which presented what is probably the most widely accepted philosophical interpretation of the technical machinery. While these developments did not dispel all the controversy, nowadays first-order modal logic together with (some form of) relational semantics, is generally regarded as a well understood (perhaps even boringly familiar) tool of philosophical analysis.

Viewed from a mathematical perspective, however, things look rather different. Had first-order modal logic never existed, a logician who proposed its (now standard) syntax and relational semantics might have been regarded as audacious, perhaps downright careless. Why? Because, in essence, first-order modal logic is a combined logic. As we have just seen, combining two modal logics while retaining interesting properties is no easy matter. So it should not come as too much of a surprise that combining propositional modal logic with first-order logic is unlikely to be plain sailing. In what follows we shall sketch the standard syntax and semantics, and mention some of its problematic features.

First the syntax (we omit some of the clauses for the booleans):

$$\varphi \quad ::= \quad P(x_1, \ldots, x_n) \mid x = y \mid \neg\varphi \mid \varphi \to \psi \mid \Diamond\varphi \mid \Box\varphi \mid \exists x\varphi \mid \forall x\varphi.$$

Here P is an n-place predicate symbol and the x_i are individual variables. So (given the clauses for the quantifiers and booleans) it is clear that we have a full first-order language at our disposal, and hence (because of the presence of the modalities) we can now search for first-order information at accessible states in the familiar way. But we can do more. The clauses for the quantifiers hide a subtlety: if a formula φ contains free first-order variables within the scope of a modality, then formulas of the form $\forall x\varphi$ and $\exists x\varphi$ bind variables within the scope of the modality. This possibility is what led to Quine's philosophical objections ("no binding into intensional contexts"). And from a technical perspective it means we are combining two very different styles of logic in a way that allows a strong form of interaction.

The standard semantics for first-order modal logic comes in a number of variant forms. One basic choice concerns the domain of quantification: should the quantifiers range over some fixed domain of quantification (the *constant domain* semantics), or should each point be associated with its own domain (the varying domain semantics)? Here we shall present the varying domain semantics; for a discussion of the constant domain approach, and of equivalences between the constant domain, varying domain, and other approaches, see Chapter 9 of this handbook, or Fitting and Mendelsohn [49].

DEFINITION 47. A varying domain model is a tuple $(W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W})$. Here W is a non-empty set; R is a binary relation on W; D (the domain of quantification) is a non-empty set; for all $w \in W$, $\delta_w \subseteq D$; and for all $w \in W$, V_w is a function that assigns to each n-place predicate symbol a subset of D^n.

That is, we have the familiar modal machinery from the propositional case (note that (W, R) is just a frame, and the V_w are essentially our familiar valuations upgraded to interpret first-order n-place predicate symbols P rather than proposition symbols p) augmented by a specification (the δ_w) of the individuals the quantifiers at each state w range over. We interpret first-order modal logic by taking such a model, together with an assignment of values to variables (that is, a function g that maps the individual variables to elements of D), and using the following satisfaction definition:

$$
\begin{array}{rcl}
\mathfrak{M}, g, w \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots g(x_n)) \in V_w(P), \\
\mathfrak{M}, g, w \models x = y & \text{iff} & g(x) = g(y), \\
\mathfrak{M}, g, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, g, w \models \varphi, \\
\mathfrak{M}, g, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, g, w \not\models \varphi \text{ or } \mathfrak{M}, g, w \models \psi, \\
\mathfrak{M}, g, w \models \Diamond\varphi & \text{iff} & \text{for some } v \in W \text{ such that } Rwv \text{ we have } \mathfrak{M}, g, v \models \varphi, \\
\mathfrak{M}, g, w \models \Box\varphi & \text{iff} & \text{for all } v \in W \text{ such that } Rwv \text{ we have } \mathfrak{M}, g, v \models \varphi, \\
\mathfrak{M}, g, w \models \exists x\varphi & \text{iff} & \text{for some } g' \sim_x g \text{ where } g'(x) \in \delta_w \text{ we have } \mathfrak{M}, g', v \models \varphi, \\
\mathfrak{M}, g, w \models \forall x\varphi & \text{iff} & \text{for all } g' \sim_x g \text{ such that } g'(x) \in \delta_w \text{ we have } \mathfrak{M}, g', v \models \varphi.
\end{array}
$$

(Here $g' \sim_x g$ means that the assignments g and g' are identical save possibly in the value they assign to the variable x.)

This language is capable of expressing some important distinctions. Consider, for example, the formulas $\forall x\Box\varphi$ and $\Box\forall x\varphi$. The first asserts, of each existing entity, that it has the property φ at all accessible states. The second asserts that, at each accessible state, each entity that exists at that particular state has property φ. Should either of these formulas imply the other? That is, should we accept as valid either of the following two principles?

$$
\begin{array}{ll}
\forall x\Box\varphi \to \Box\forall x\varphi & \text{Barcan formula} \\
\Box\forall x\varphi \to \forall x\Box\varphi & \text{Converse Barcan formula}
\end{array}
$$

Instead of trying to answer such tricky philosophical questions (which bear on the *de dicto*/*de re* distinction, discussed in Chapter 9 of this handbook) let us consider what they say in the light of the relational interpretation just given. It is not difficult to see that the Barcan formula is valid in a varying domain model iff that model has *decreasing domains*, that is, if for all $w, v \in W$, Rwv implies $\delta_v \subseteq \delta_w$. And the Converse Barcan formula is valid on precisely *increasing domain* models, that is, models with the property that Rwv implies $\delta_w \subseteq \delta_v$. So to insist on the validity of both principles is to force an even stronger interaction between the quantifiers and modalities: it takes us to a locally constant domain semantics in which Rwv implies $\delta_w = \delta_v$. This is a good example of the clarity that relational semantics can bring to difficult conceptual issues, and shows why first-order modal logic can be useful in philosophical logic and natural language semantics.

So what's the problem? Simply this: for all its analytical utility, first-order modal logic under its standard semantics is not well behaved mathematically. Early signs of trouble appeared in Fine [45], which showed that interpolation and the Beth property fail for first-order **S5** under the varying domain semantics, and for any first-order modal logic between **K** and **S5** under the constant domain semantics. As **S5** is both philosophically central (it is widely considered to embody the logic of "necessarily" and "possibly") and semantically straightforward (it is the logic of frames in which R is an equivalence relation) these are strong negative results indeed. Worse was to come. It turns out that it is possible to take a propositional modal logic that is complete with respect to some class of frames, axiomatically extend it in the manner naturally suggested by the standard semantics, and yet to wind up with an incomplete first-order modal logic (see Ghilardi [56], Shehtman and Skvortsov [117], Corsi and Ghilardi [26], Cresswell [27]).

Now, the issue here is not so much the incompleteness in itself (as we have already discussed, even in the propositional modal logic, frame incompleteness results are the norm) rather it is the *loss* of completeness in the transition from the propositional case to the first-order case that is worrying. To use the terminology introduced when we discussed combinations of logics: the standard relational semantics for first-order logic is a method of combination for which transfer of completeness fails.

Such results have led to renewed technical interest in first-order modal logic. The semantics of first-order modal logic has come under intense scrutiny, and a number of alternative semantics have been proposed which enable completeness results to be transferred. Some of this work has been model-theoretic (see, in particular, van Benthem's [132] use of functional frames) but most of it has been highly abstract, employing the language of category theory; for a detailed account of such work, see Chapter 9 of this handbook. More recently, the hybrid logic community has pointed out that upgrading the underlying propositional modal language to a hybrid language is another way to repair the situation: interpolation is regained (see Areces, Blackburn and Marx [7]), indeed, regained constructively (see Blackburn and Marx [15]) and general positive results on transfer of completeness can be proved (see Blackburn and Marx [14]). All in all, first-order modal logic is one of the most intriguing areas of modal logic: the most venerable system of all poses some of the most interesting questions about what it is to be modal.

6.10 General perspectives

Moving to richer languages better fitted for particular applications is a standard feature of current research. It is true that in some quarters sticking to the poorest modal base language of the founding fathers (despite its evident handicaps in expressive power and mathematical convenience) is still something of a religion. But the idea of designing extensions is not some new-fangled notion; its roots stretch back to the work of von Wright [145] and Prior [104, 105], and the idea was central to the work of the Sofia School (see, for example, Passy and Tinchev [101] for insightful comments on what modal logic is and why one might want to enrich it). Still, pointing to a noble heritage is not enough. We need to address a tricky question: what makes these languages *modal*? Being precise here is difficult. As we have seen, there is a wide range of extensions. Moreover, each application imposes its own concerns and peculiarities. Nevertheless, there is a guiding idea that lies behind most examples of this form of language design: obtaining a reasonable balance between expressive power and computational complexity. So the question we should focus on is: what makes such natural balances arise?

As we have seen, many richer modal languages are fragments of the full language of first-order logic over some appropriate similarity type of relations and properties. We can see this by translation, just as we did with the basic modal language (we saw that the complex truth conditions for the Until and Since are definable by first-order formulas, and the same is true for the conditional connective, the universal modality, and the apparatus of hybrid logic). Now, there have been various attempts to find general patterns explaining which parts of first-order logic are involved in modal languages. Gabbay [51] observed that modal languages tend to translate into so-called *finite variable fragments* of first-order logics, that is, fragments using only some finite number of variables, fixed or bound. For example, we have seen that the basic modal language can make do with only two variables, and temporal logic with Until and Since, and conditional logic, only require three. Finite variable fragments have some pleasant computational behaviour; for example, their model checking complexity is in PTIME (see Vardi [141]) as opposed to PSPACE for the full first-order language. On the other hand, as we have already mentioned, satisfiability is already undecidable for first-order fragments with three variables, so the real reason for the low complexity of modal languages lies elsewhere. A different type of analysis for the latter phenomenon was given in the paper "Why is modal logic so robustly decidable?" (Vardi [142]). This emphasises the semantic adequacy of the tree-like models obtainable via bisimulation unraveling of arbitrary graph models. This type of explanation is important as it transcends first-order logic; on the other hand it does not provide much in the way of concrete syntactic insight. For the latter, the current best explanation is the one provided by the guarded fragment and its relatives (which are, arguably, the strongest known modal languages).

As we saw, guarded fragments locate the essence of modal logic in the *restriction* on the quantification performed by the modalities. One attractive property of this analysis is its logical resilience: it turns out that it extends beyond the setting of first-order enrichments to second-order enrichment too, something that was not forseen when the guarded fragment was first isolated. A striking example is the result in Grädel and Walukiewicz [63] that the extension of the guarded fragment with the fixed-point operators μ and ν remains decidable. By way of contrast, validity for full first-order logic extended with these operators is non-axiomatisable, indeed, non-arithmetical. This observation shows that the modal philosophy embodied in the idea of guarded fragments is not restricted to first-order extensions: often modal fragments can bear the weight of additional higher-order apparatus (such as fixed-point operators) which would send the full first-order correspondence languages into a tailspin complexity wise. Our discussion of PDL and the modal μ-calculus has shown that this is the case for the basic modal language. Grädel and Walukiewicz's result for the guarded fragment shows that this type of behaviour persists higher up: guarded quantification can support higher-order constructions too.

Perhaps guarding can be a fruitful strategy in even more exotic modal settings? One setting worth exploring is *infinitary* modal logic. This logic (which was used extensively in Barwise and Moss [10] and Baltag [8] for investigating non-well founded set theory; see Chapter 16 of this handbook) provides a perfect match with bisimulation: two pointed models are bisimilar if and only if they satisfy the same formulas in a modal language that allows arbitrary infinite conjunctions and disjunctions. Moreover a modal characterisation theorem holds. Now, decidability is a non-issue in this setting, but what about existential semantic properties such as interpolation and Beth Definability? It is known that interpolation holds for infinitary modal logic (see Barwise and van Benthem [11]), but can such results be lifted to infinitary guarded fragments? Another setting worth exploring in this way is *second-order propositional modal logic*, in which we can quantify over proposition symbols (see Fine [42] for some early results, ten Cate [124] for a more recent discussion, and Chapter 10 of this handbook for a brief overview). The equation "modality = guarding" should be simultaneously regarded as a hypothesis to be tested in richer settings, and as a useful heuristic for isolating further logics worth calling modal.

Not that we should put all our eggs in one basket. Perhaps the notion of modality is too diffuse for any single approach to exhaust, and in any case it is worth looking for alternatives. Another approach is to apply ideas from abstract model theory (see Barwise and Feferman [9]). This was first done in de Rijke [30], who proved a modal analog of Lindström's [91] celebrated characterisation of first-order logic. The original form of Lindström's theorem says that an abstract logic $\mathcal{L}$ extending first-order logic coincides with first-order logic iff it has the compactness and Löwenheim-Skolem properties. Another way of stating the theorem is that an abstract logic $\mathcal{L}$ extending first-order logic coincides with first-order logic iff it has the compactness and Karp properties. (The Karp property is that all formulas are invariant for potential isomorphism, where a potential isomorphism is a non-empty family of finite partial isomorphisms closed under the usual back and forth extension properties; recall our discussion of partial isomorphisms in Section 3.3). We shall discuss a (slightly reformulated) version of de Rijkes's result and a more recent characterisation due to van Benthem.

What is an abstract modal logic? Here's the conception that underlies our reformulation of de Rijke's result. We give it in terms of pointed models $(\mathfrak{M}, w)$, that is, a model together with a point of evaluation.

DEFINITION 48 (Very abstract modal logics). Let $\mathcal{L}$ be a set of formulas, and $\models_{\mathcal{L}}$ its satisfaction relation, that is, a relation between pointed models and $\mathcal{L}$-formulas. A very abstract modal logic is a pair $(\mathcal{L}, \models_{\mathcal{L}})$ with the following properties:

1. Occurrence property. For each φ in $\mathcal{L}$ there is an associated finite language $\mathcal{L}(\lambda_\varphi)$. The relation $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ is a relation between $\mathcal{L}$-formulas φ and models $(\mathfrak{M}, w)$ for languages $\mathcal{L}$ containing $\mathcal{L}(\lambda_\varphi)$. That is, if φ is in $\mathcal{L}$, and $\mathfrak{M}$ is an $\mathcal{L}$-model, then $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ is either true or false if $\mathcal{L}(\lambda_\varphi) \subseteq \mathcal{L}$, and undefined otherwise.

2. Expansion property. The relation $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ depends only on the restriction of $\mathfrak{M}$ to $\mathcal{L}(\lambda_\varphi)$. That is, if $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ and $(\mathfrak{N}, w)$ is an expansion of $(\mathfrak{M}, w)$ to a larger language, then $(\mathfrak{N}, v) \models_{\mathcal{L}} \varphi$.

A very abstract modal logic $(\mathcal{L}, \models_{\mathcal{L}})$ extends basic modal logic if for every basic modal formula there exists an equivalent $\mathcal{L}$-formula (that is, if for each basic modal formula φ there exists an $\mathcal{L}$-formula ψ such that for any model $(\mathfrak{M}, w)$ we have $(\mathfrak{M}, w) \models \varphi$ iff $(\mathfrak{M}, w) \models_{\mathcal{L}} \psi$).

De Rijke's characterisation centres on the familiar bisimulation invariance property and the *finite depth property*. A very abstract modal logic $\mathcal{L}$ has the *finite depth property* iff for any $\mathcal{L}$-formula φ there is some natural number k such that for all models $\mathfrak{M}$,

$$\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{M}|k, w \models \varphi,$$

where $\mathfrak{M}|k$ is the model $\mathfrak{M}$ restricted to just those points that can be reached from w in k or fewer R-steps. De Rijke builds invariance for bisimulation into the notion of abstract modal logic, so his statement of his Lindström-style result has the form: any abstract modal logic with the finite depth property that extends the basic modal language is the basic modal language. Reformulating his result in terms of very abstract modal logics, thereby making the bisimulation invariance condition explicit, results in:

THEOREM 49. *Suppose $\mathcal{L}$ is a very abstract modal logic extending the basic modal language. Then $\mathcal{L}$ coincides with the basic modal language iff $\mathcal{L}$ has the finite depth and invariance for bisimulation properties.*

Proof. See de Rijke [30, 31]. For a textbook-level exposition of the proof, see Theorem 7.60 of Blackburn, de Rijke and Venema [13]. ■

This is an informative result. Nonetheless, the finite depth property seems somewhat engineered to capture the basic modal language, and it is natural to look for generalisations. However, because of the expressive limitations of modal languages, this is not straightforward. The proof of the Lindström Theorem for first-order logic typically proceeds by contradiction: to show that an abstract first-order formula has a first-order equivalent, one typically builds a model where φ is true in one part, $\neg\varphi$ in another, and uses the expressive power of first-order logic to link the two parts of the model by a chain of partial isomorphisms, thereby reaping the contradiction. This style of argument does not lift easily to modal languages: the basic modal language is too impoverished to encode the chains of bisimulations linking the two parts of the model that would be required to mimic this proof technique directly. However, as van Benthem [139] observed, there is a way around this. The key idea is to strengthen the definition of a very abstract modal language by demanding it fulfils the *relativisation* condition:

DEFINITION 50 (Abstract modal logics). An abstract modal logic $\mathcal{L}$ is a very abstract modal logic that has the *relativisation property*: for any $\mathcal{L}$-formula φ and proposition symbol p not occurring in φ, there is a formula *Rel*(φ, p) which is true at a model $(\mathfrak{M}, w)$ iff φ is true at $(\mathfrak{M}|p, w)$, which is the submodel of $\mathfrak{M}$ consisting of just those points that satisfy p.

Relativisation is a natural property (most logics satisfy it) but the key point is to observe is how it is used in the proof of the following theorem: in essence, it provides a model-theoretic tool which enables us to give an alternative proof without resorting to explicit codings of bisimulations. This leads to van Benthem's version of the Lindström Theorem for modal logic:

THEOREM 51. *Suppose $\mathcal{L}$ is an abstract modal logic extending the basic modal language. Then $\mathcal{L}$ coincides with the basic modal language iff $\mathcal{L}$ satisfies compactness and invariance for bisimulation.*

Proof. We know that the basic modal language satisfies compactness (Proposition 4) and invariance for bisimulation (Lemma 9) so the left to right direction is clear. For the reverse direction, assume that $\mathcal{L}$ has these properties. We claim that the following holds: *in a compact abstract modal logic $\mathcal{L}$ which is invariant for bisimulations, every formula has the finite depth property.* If we can show this, the result follows from Theorem 49.

We prove the claim as follows. Let φ be any formula in $\mathcal{L}$. Suppose for the sake of a contradiction that φ lacks the finite depth property. Then for any natural number k there exists a model $(\mathfrak{M}_k, w)$ and a cut-off version $(\mathfrak{M}_k|k, w)$ which disagree on the truth value of φ. Without loss of generality, assume that the following happens for arbitrarily large k: $(\mathfrak{M}_k|k, w) \models \varphi$, and $(\mathfrak{M}_k, w) \models \neg\varphi$ (here we use the fact that abstract modal logics are closed under negation). Now take a new proposition symbol p, and consider the following set Σ of $\mathcal{L}$-formulas:

$$\{\neg\varphi, \textit{Rel}(\varphi, p)\} \cup \{\Box^n p \mid \text{for all natural numbers } n\}.$$

(By $\Box^n p$ we mean p prefixed by a sequence of n boxes.) Given our assumptions, this set is finitely satisfiable: we choose k sufficiently large, and make p true in the k reachable part of one of the above sequences of models. But then, by compactness for our abstract modal logic $\mathcal{L}$, there must be a model $(\mathfrak{N}, v)$ for the whole set Σ at once.

But this leads to a contradiction as follows. We focus on the generated submodel $(\mathfrak{N}_v, v)$ consisting of v and all points finitely reachable from it. Now, the identity relation is a bisimulation between any pointed model and its unique generated submodel. Hence, by the assumed invariance for bisimulation, formulas of $\mathcal{L}$ have the same truth value in any pointed model and its generated submodel. Now, given our definition of Σ, $\neg\varphi$ holds in $(\mathfrak{N}, v)$, and hence also in $(\mathfrak{N}_v, v)$. On the other hand, since $(\mathfrak{N}, v) \models$ *Rel*(φ, p), we have $(\mathfrak{N}|p, v) \models \varphi$. But by the truth of all the formulas of the form $\Box^n p$, p holds in the whole generated submodel $(\mathfrak{N}_v, v)$. Therefore we have that φ holds in (N_v, v). Contradiction. Hence the claim is established and the theorem follows. ■

One surprising consequence of this result is that the Modal Characterisation Theorem (Theorem 13) follows from it; see van Benthem [139] for details.

It remains to be seen how widely applicable this technique is. For example, it is not straightforwardly applicable to languages with the universal modality, as these lack the finite depth property. However it *can* be lifted to the guarded fragment. As we mentioned in Section 6.5, there is a notion of guarded bisimulation. And using this notion, together with the relativisation technique leads to:

THEOREM 52. *Suppose $\mathcal{L}$ is an abstract modal logic extending the guarded fragment. Then $\mathcal{L}$ coincides with the guarded fragment iff $\mathcal{L}$ satisfies compactness and invariance for guarded bisimulation.*

Proof. See van Benthem [139]. ■

7 ALTERNATIVE SEMANTICS

As we said at the start of this chapter, one of the most instructive ways of thinking about modal logic is to view it as a tool for talking about graphs. But to view modal logic exclusively through the lens of relational semantics would be a mistake; interesting alternatives exist, and in this section we introduce three of them: algebraic semantics, neighbourhood semantics, and topological semantics. As we shall see, each of these semantics has something new to offer. But we shall come across much that is familiar, for all three are linked in various ways with relational semantics.

7.1 Algebraic semantics

The basic idea of algebraic semantics is simple: view modal formulas as terms (or polynomials) and evaluate them in the appropriate type of algebra. So the key question is: what kinds of algebra are appropriate for modal logic? The answer is: *boolean algebras with operators*, or BAOs.

A *boolean algebra* is a triple $\mathfrak{A} = (A, +, \times, -, 1, 0)$ such that both $+$ (*join*) and $\times$ (*meet*) are commutative and associative binary operations, each of which distributes over the other. The unary operation $-$ (*complement*) must satisfy the equations $x + (-x) = 1$ and $x \times (-x) = 0$. The nullary operations (or *constants*) 1 and 0 must satisfy the equations $x \times 1 = x$ and $x + 0 = x$.

Even if you have never encountered boolean algebras before, a moment's reflection should make it clear that they are an algebraic mirror of propositional logic. To see this, read $+$ as $\vee$, $\times$ as $\wedge$, $-$ as $\neg$, 1 as $\top$, 0 as $\bot$, and $=$ as $\leftrightarrow$. So it only remains to provide algebraic structure that mirrors the diamonds. This motivates the following definition.

DEFINITION 53 (Boolean Algebras with Operators). A boolean algebra with operators, or BAO, is a pair $\mathfrak{B} = (\mathfrak{A}, m)$, where $\mathfrak{A}$ is a boolean algebra and m is a unary operator on $\mathfrak{A}$ that satisfies the equations $m(x + y) = m(x) + m(y)$, and $m(0) = 0$.

Note that the logical analogs of these two equations are $\Diamond(\varphi \vee \psi) \leftrightarrow (\Diamond\varphi \vee \Diamond\psi)$, and $\Diamond\bot \leftrightarrow \bot$, both of which are valid in relational semantics. Thus we now have an algebraic mirror for all components of the basic modal language.

We interpret the basic modal language in BAOs in the usual algebraic fashion. That is, given a BAO, we view the proposition symbols as variables ranging across the elements of the algebra, and interpret each logical operator by its corresponding algebraic operation. More precisely, let $\mathfrak{B}$ be a BAO, and V be a function mapping each proposition symbol to an element of $\mathfrak{B}$; we call such a function V an *algebraic valuation*. We extend V to a function that gives the result of evaluating arbitrary basic modal formulas in $\mathfrak{B}$ via the following recursive clauses:

$$\begin{array}{rcl} V(\varphi \vee \psi) & = & V(\varphi) + V(\psi) \\ V(\varphi \wedge \psi) & = & V(\varphi) \times V(\psi) \\ V(\neg\varphi) & = & -V(\varphi) \\ V(\Diamond\varphi) & = & mV(\varphi) \end{array}$$

It is now possible to prove the following algebraic completeness result:

THEOREM 54. *A basic modal formula belongs to the minimal modal logic **K** iff it evaluates to the value* 1 *in all modal algebras under all algebraic valuations.*

Proof. Straightforward. The key point is to use a technique standard in algebraic logic, namely to create the *Lindenbaum-Tarski Algebra* for **K**. The elements of the Lindenbaum-Tarski Algebra are equivalence classes of **K**-provably equivalent formulas; the operations are defined with the aid of the connectives. All and only the **K**-provable formulas evaluate to 1 in this algebra, and hence the result follows. For a detailed discussion, see Chapter 6 of this handbook. ■

In fact, a far stronger result can be proved: *any* axiomatic extension of **K** (that is, *any* normal modal logic) is complete with respect to some class of algebras. And the proof is not difficult. In essence, one replicates the completeness proof for **K**, but works with the Lindenbaum-Tarski Algebra which satisfies the additional axiomatic constraints. As we saw earlier (recall Theorem 26) there is no general completeness result for normal modal logics with respect to frames. This is an important difference between algebraic and relational semantics.

Nonetheless, it is likely that some readers will feel a little cheated. Isn't the whole approach really just syntax in disguise? After all, algebraic semantics matches the modal language with algebraic operations that transparently mirror fundamental validities of the original logic. This does not seem like genuine semantic analysis: it has more the flavour of linking two distinct, but closely related, syntactic realms. Moreover, the algebraic satisfaction definition has a global rather than a local flavour.

This is true, but somewhat besides the point, for in spite of the general completeness result just noted, we have not yet entered the heartland of algebraic semantics. For what algebraic semantics really provides is a doorway to a larger mathematical universe. The power of algebraic semantics comes from the wealth of ideas and techniques it enables us to bring to bear on problems in modal logic. Some of these techniques take us back, via a novel path, to the heart of relational semantics, but others take us to new territory. Let's look a little deeper.

An important theme in algebra is the *representation* of abstract mathematical structures by concrete set-theoretic structures. The point of a representation theorem is to show that some abstractly specified class of algebras picks out an intended class of concrete structures. So representation theorems are rather like completeness theorems: they show that the abstract (often equational) specification is strong enough to ensure that every abstract algebra is isomorphic to a concrete algebra. Two classic examples are Cayley's Theorem, which shows that every finite group is isomorphic to a collection of permutations, and the Stone Representation Theorem, which shows that every abstract boolean algebra is isomorphic to a field of sets (that is, a boolean closed collection of subsets of some W that contains W) with $\times$ viewed as intersection, $+$ viewed as union, and $-$ viewed as set-theoretic complement. Now, in 1952, several years before relational semantics was officially invented, Jónsson and Tarski [74, 75] proved a remarkable representation theorem for BAOs: they showed that every abstract BAO could be represented as a relational structure. Inexplicably, their paper made no mention of modal logic. This was unfortunate as their paper contained all the technical machinery needed to define *relational* semantics and prove *relational* completeness results for most commonly occurring modal logics. In essence, their result allows relational completeness proofs to be factored into an algebraic completeness step (which makes use of the Lindenbaum-Tarski Algebra) followed by a representation step (which turns this algebra into a relational structure). Nowadays, the Jónsson-Tarski Theorem is rightly considered a cornerstone of modal logic; for a detailed proof of the theorem, and examples of how to put it to work, see Chapter 6 of this handbook.

Another important theme goes under the name of *duality theory*. As we saw in Section 5, there are four key transformations on frames (disjoint unions, generated submodels, bounded morphisms, and ultrafilter extensions) and, as the Goldblatt-Thomason Theorem tells us, closure of a frame class under these model-theoretic constructions is necessary and sufficient to ensure its basic modal definability. But as we have already remarked (see Theorem 33) the original proof of the Theorem was *algebraic*. What's the algebraic connection? This: each of these four operations on frames corresponds to an operation on classes of algebras. Viewed this way, the Goldblatt-Thomason Theorem can be seen as a modal version of the Birkhoff Theorem, which identifies equationally definable classes of algebras with those classes of algebras that are closed under the formation of subalgebras, homomorphisms, and products. For a detailed discussion, we again refer the reader to Chapter 6.

But important as these two examples are, they merely hint at the wealth of techniques made available by the algebraic connection. Algebraic semantics has repeatedly proved itself a powerful analytical tool. To give another classic example, Blok [16] was able to give a detailed analysis of frame incompleteness by drawing on algebraic methods. In particular, he did so by investigating *splittings* (a concept from lattice theory) of the lattice of normal modal logics; for a discussion of Blok's work, see Chapter 7 of this handbook. Moreover, in many cases algebraic methods have been adapted to richer modal languages. A nice example is provided by the universal modality. In the algebraic setting, the universal modality allows us to define a *discriminator term*, that is, a term denoting an operator that maps 0 to 0 and all other elements to 1. Algebras with discriminator terms are particularly straightforward to work with (see Chapter 6 of this handbook) thus here algebraic semantics sheds interesting light on a relationally-natural extension of the basic modal language. But algebraic semantics also illuminates areas where relational semantics has little to say. For example, it turns out that the boolean structure of the underlying algebras is not particularly significant. That is, it is possible to analyse modalities algebraically even if we *don't* have full classical propositional logic at our disposal. Such logics can be important in various settings, and relational semantics at present offers little in the way of insight. For further remarks and references on this application of algebraic semantics, see Chapter 6 of this handbook.

7.2 Neighbourhood semantics

For some applications, relational semantics is too strong. For example, $\Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi)$ is valid under relational semantics. But if we read $\Diamond\varphi$ as making the game-theoretic assertion that the player has a strategy forcing the outcome to satisfy φ, we might be inclined to reject it: why should possession of a strategy for a disjunction imply possession of a strategy for one of the disjuncts? For example, suppose we play a game with the following moves: you have the right to decide whether we go to a movie or a concert, and I can decide which particular movie or concert we go to. Suppose the movie I want to see is *Crash*, and that my favourite music is *Mozart*. It follows that I can force *Crash* $\vee$ *Mozart*, but (because it's you who determines the movie/concert option) I can't determine which of these two options will actually take place. Similarly, if we interpret $\Box\varphi$ epistemically we have further grounds for objection. For a start, relational semantics validates the following principle:

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi).$$

Moreover, it validates the following pattern of inference: if $\models \varphi$ then $\models \Box\varphi$. These work together to enforce a strong form of logical omniscience: if an agent knows φ, then she knows all its logical consequences.

Such considerations have led to a search for weaker semantics. Perhaps the best known of these is neighbourhood semantics (introduced in Montague [96, 97] and Scott [112] and explored in Segerberg [113]). The key idea of neighbourhood semantics has a topological flavour: each point w in a model is associated with a collection of subsets of the domain (the neighbourhoods of w) and a formula of the form $\Box\varphi$ is true at w iff the set of points in a model satisfying φ is a neighbourhood of w. Let's make this precise. A neighbourhood model is a triple (W, R, V) where W is a non-empty set of states, V is a valuation, and R relates points $w \in W$ to subsets of W (that is, $R \subseteq W \times 2^W$). For any $w \in W$, let N_w be $\{U \subseteq W \mid wRU\}$; we call N_w the set of neighbourhoods of w. We interpret boxed formulas as follows:

$$\mathfrak{M}, w \models \Box\varphi \text{ iff } \{u \in W \mid \mathfrak{M}, u \models \varphi\} \in N_w,$$

and use the dual definition for diamonds:

$$\mathfrak{M}, w \models \Diamond\varphi \text{ iff } \{u \in W \mid \mathfrak{M}, u \not\models \varphi\} \notin N_w.$$

Neighbourhood semantics is a generalisation of relational semantics. To see this, note that given any relational model $\mathfrak{M} = (W, R, V)$ we can form a neighbourhood model $\mathfrak{M}^n = (W, R^n, V)$ by stipulating, for each $w \in W$ and $U \subseteq W$, that $R^n wU$ iff $U = \{u \in W \mid Rwu\}$. That is, for each $w \in W$, N_w is the singleton set containing the set of points that are R-accessible from w. Hence, for all $w \in W$ and all basic modal formulas φ, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}^n, w \models \varphi$. In short, we can turn any relational model into an equivalent neighbourhood model.

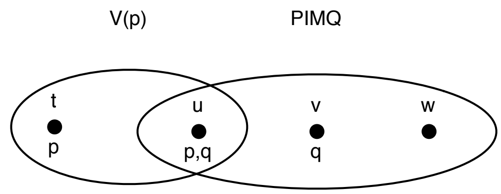

Figure 22. Neighbourhood model that falsifies $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ at u.

But we cannot do the reverse. Consider a model $\mathfrak{M} = (W, R, V)$ such that $W = \{t, u, v, w\}$, $V(p) = \{t, u\}$ and $V(q) = \{u, v\}$, and $N_u = \{V(p), \textit{PIMQ}\}$, where *PIMQ* $= \{u, v, w\}$. Such a model is shown in Figure 22; note that *PIMQ* is the set of points where $p \to q$ is true. Hence $\mathfrak{M}, u \models \Box(p \to q)$, as *PIMQ* $\in N_u$. Furthermore, $\mathfrak{M}, u \models \Box p$, as $V(p) \in N_u$. However $\mathfrak{M}, u \not\models \Box q$, for $V(q) \notin N_u$. So $\mathfrak{M}, u \not\models \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$. As this formula is valid under relational semantics, no relational model equivalent to $\mathfrak{M}$ exists.

Moreover, the inferential principle characteristic of relational semantics (if $\models \varphi$ then $\models \Box\varphi$) no longer holds. To see this, it suffices to consider a model $\mathfrak{M}$ consisting of a single point w such that $N_w = \emptyset$. Then $\mathfrak{M}, w \models \top$, but $\mathfrak{M}, w \not\models \Box\top$. In fact, all that remains in neighbourhood semantics is the weaker principle: if $\models \varphi \leftrightarrow \psi$ then $\models \Box\varphi \leftrightarrow \Box\psi$. Thus neighbourhood semantics does not enforce logical omniscience.

Neighbourhood semantics has been criticised as under-motivated. It may banish the spectre of logical omniscience, but does it do so in a principled way? After all, isn't there something stipulative, indeed ad-hoc, about simply asserting that certain subsets and not others are in the neighbourhood of a given point? There is a grain of truth in such criticisms, nonetheless we should not be too quick to dismiss the approach. For some applications, asserting that certain neighbouring regions are important is probably the best we can do in the way of semantic analysis. Furthermore, like relational semantics, neighbourhood semantics offers an entire *framework* for semantics; imposing further restrictions on neighbourhoods (for example, demanding that neighbourhoods be superset closed) is a mechanism which permits finer-grained semantic analyses to be attempted. See Chellas [24] for an introduction to some of the options here.

Neighbourhood semantics has some pleasant properties. For a start (if NP $\neq PSPACE$, the standard assumption) it is better behaved computationally than relational semantics:

THEOREM 55. *The satisfiability problem for neighbourhood semantics is NP-complete.*

Proof. See Vardi [140]. The key observation is that if a formula φ is satisfiable in a neighbourhood model, then it is satisfied in a model with at most $|\varphi|^2$ states, where $|\varphi|$ is the number of symbols in φ. ■

Moreover, neighbourhood semantics meshes well with the algebraic and coalgebraic approaches discussed in Chapter 6 of this handbook.

7.3 Topological semantics

Topological semantics is one of the oldest modal semantics, and the first in which deep technical results were proved. In 1938, Tarski [123] showed that **S4** (the logic which in relational semantics is complete with respect to transitive and reflexive frames) is complete with respect to topological spaces. Then, in 1944, McKinsey and Tarski [95] showed that **S4** is the modal logic of the real numbers, and indeed of any metric separable space without isolated points. Since this pioneering work, topological semantics has been deeply (if somewhat sporadically) studied, and many interesting results have been proved (see for example Esakia [38] and Shehtman [115]) but for many years it was rather isolated from the modal mainstream. More recently, however, partly because of the growing interest in logics of space, there has been a revival of interest. For an overview of developments in topological semantics since the time of Tarski, see Chapter 16 of this handbook; here we will introduce its basic ideas in a way that emphasises connections with our account of relational semantics. Our discussion is based on Aiello, van Benthem, and Bezhanishvili [2].

A *topological space* is a pair (W, τ), where W (the *domain*) is a non-empty set and τ (the *topology*) is a collection of subsets of W that contains both $\emptyset$ and W, is closed under finite intersections (that is, if $O, O' \in \tau$ then $O \cap O' \in \tau$) and closed under arbitrary unions (if $\{O_i\}_{i\in I} \in \tau$ then $\bigcup_{i\in I} O_i \in \tau$). A topology τ such that $\tau = 2^W$ is called *discrete*, and a topology such that $\tau = \{\emptyset, W\}$ is called *trivial*. If (W, τ) is a topological space and $O \in \tau$ then O is called an *open set*. If w is a point in an open set O, then O is called an *open neighbourhood* of w. A *closed set* is the complement of an open set.

A *topological model* is a triple $\mathfrak{M} = (W, \tau, V)$ where (W, τ) is a topological space and V is a valuation (in the sense familiar from relational semantics). We interpret proposition symbols and booleans in the usual way, but what about the modalities? Boxed formulas are handled as follows:

$$\mathfrak{M}, w \models \Box\varphi \text{ iff } (\exists O \in \tau)(w \in O \text{ and } (\forall u \in O)(\mathfrak{M}, u \models \varphi)).$$

That is, $\Box\varphi$ is true at w iff it is true at all the points of some open neighbourhood of w. Diamonds are handled dually:

$$\mathfrak{M}, w \models \Diamond\varphi \text{ iff } (\forall O \in \tau)(w \in O \text{ implies } (\exists u \in O)(\mathfrak{M}, u \models \varphi)).$$

That is, $\Diamond\varphi$ is true at w iff it is true at some point in each open neighbourhood of w.

At first blush, this looks very different from relational semantics. And there *are* some obvious semantic differences. For example, the characteristic axioms of **S4**, namely $\Box p \to p$ and $\Box p \to \Box\Box p$, are valid on all topological models, so the minimal logic is stronger than in relational semantics. But a closer look reveals the similarities. For a start, like relational semantics, topological semantics is local: the truth value of a formula at a point only depends on what happens inside the open neighbourhoods of that point. More precisely, suppose that w is a point in a topological model $\mathfrak{M}$, and that O is an open neighbourhood of w. Let $\mathfrak{M}|O$ be the model with domain O whose open sets are all the open subsets of O in $\mathfrak{M}$, and whose valuation is the restriction of the valuation V of $\mathfrak{M}$ to O (that is $V|O(p) = V(p) \cap O$). Then a simple induction shows that for all basic modal formula φ, and all points $w \in O$, $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}|O, w \models \varphi$. Nor is it hard to find other similarities. For example, the fact that **S4** has the finite model property with respect to relational semantics is neatly matched by the fact that the basic modal language has the finite model property with respect to topological semantics.

But the similarities run deeper than these examples might suggest. In particular, topological semantics gives rise to a natural notion of bisimulation:

DEFINITION 56 (Topo-bisimulation). A topo-bisimulation between two topological models $\mathfrak{M} = (W, \tau, V)$ and $\mathfrak{M}' = (W', \tau', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols,

Zig: if $w \in O \in \tau$, then there is an open set $O' \in \tau'$ such that $w' \in O'$ and $(\forall u' \in O')(\exists u \in O)(uEu')$, and

Zag: if $w' \in O' \in \tau'$, then there is an open set $O \in \tau$ such that $w \in O$ and $(\forall u \in O)(\exists u' \in O')(uEu')$.

If there is a topo-bisimulation between two topological models $\mathfrak{M}$ and $\mathfrak{N}$, then we say that $\mathfrak{M}$ and $\mathfrak{N}$ are topo-bisimilar. Moreover, we say that two states are topo-bisimilar if they are related by some topo-bisimulation.

Let's restate the zig clause informally: it says that for two points w and w' to be topo-bisimilar, then for any open neighbourhood O of w it must be possible to find an open neighbourhood O' of w' such that every point u' in O' is topo-bisimilar to some u in O. Figure 23 illustrates this idea (the dotted line connecting u and u' needs to be interpreted universally: *every* u' is linked to some u).

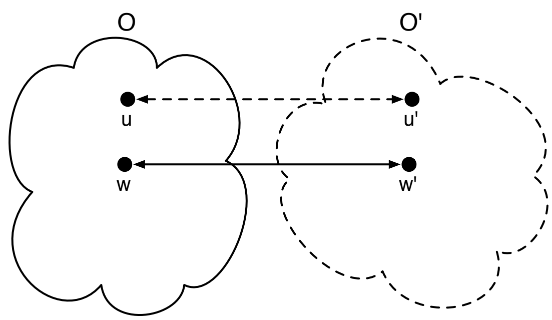

Figure 23. Zig (and zag) for topo-bisimulations

Such bisimulations are topologically natural. Two basic concepts of topology are *open maps* and *continuous maps*. For any topological spaces (W, τ) and (W', τ'), a function f from W to W' is called open if for all $O \in \tau$ we have that $f(O) \in \tau'$, and it is called continuous if for all $O' \in \tau'$ we have that $f^{-1}(O') \in \tau$. It is easy to see that every open and continuous map induces topo-bisimulations: given a valuation on one space, take its image in the other, and the resulting models are topo-bisimilar. But topo-bisimulations are also modally natural. For a start, we have the following analog of Lemma 9:

LEMMA 57 (Topo-bisimulation Invariance Lemma). *If E is a topo-bisimulation between $\mathfrak{M} = (W, \tau, V)$ and $\mathfrak{M}' = (W', \tau', V')$, and wEw', then w and w' satisfy the same basic modal formulas.*

Proof. A routine induction. ■

As a simple illustration, we noted above that $\mathfrak{M}$ and $\mathfrak{M}|O$ (the localisation of $\mathfrak{M}$ to some open set O) were equivalent. But this is unsurprising. The identity relation between the domains of the two models is a topo-bisimulation, hence the result is a special case of this lemma.

What about the converse? Characterisation results for the general case are tricky to state (we would need to discuss what a suitable correspondence language for topological semantics is, and this would take us too far afield). But we *do* have an analog of Proposition 11:

PROPOSITION 58. *If points w and w' from two finite topological models $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same modal formulas, then there is a topo-bisimulation E between $\mathfrak{M}$ and $\mathfrak{N}$ such that wEw'.*

So far so good. But just how expressive is the basic modal language in the new setting? To pose the question a little more forcefully: what (interesting) topological conditions can the basic modal language enforce via the concept of validity? Here's one example. The formula

$$p \leftrightarrow \Box p$$

is valid on a topological model iff that model bears the discrete topology (that is, iff every subset of the domain is open). This is pleasant, but many fundamental properties lie beyond the reach of the basic language. For example, a topological space (W, τ) is *connected* iff the only elements of τ that are both open and closed are W and $\emptyset$. But this condition is not basic modal definable. For suppose for the sake of a contradiction that some formula φ does define connectedness. Consider the topological space with domain $\{1, 2\}$ under the discrete topology; this space is *not* connected as $\{1\}$ and $\{2\}$ are both open and closed. Hence we can define a model $\mathfrak{M}$ on this space that will falsify φ at some point, say 1. But then $\mathfrak{M}|\{1\}$ will falsify φ at 1 too, as $\mathfrak{M}$ and $\mathfrak{M}|\{1\}$ are topo-bisimilar. But $\mathfrak{M}|\{1\}$ bears the trivial topology, hence it is a connected space, so it should validate φ. We conclude that connectedness is undefinable.

All in all, the basic modal language turns out to be disappointingly weak when it comes to standard topological conditions. But then why stick with the basic modal language? As readers of this chapter are well aware, there are interesting ways of augmenting modal expressivity, and recently these have begun to be explored in the topological setting. For example, Shehtman [116] and Aiello and van Benthem [1] observe that connectivity becomes definable when the universal modality is added to the language:

$$A(\Diamond p \to \Box p) \to (Ap \vee A\neg p).$$

And Gabelaia notes that the T_0 condition (for any two points x and y there exists either an open neighbourhood O_x of x such that $y \notin O_x$ or an open neighbourhood O_y of y such that $x \notin O_y$) is definable in the basic hybrid language by

$$@_i\neg j \to (@_j\Box\neg i \vee @_i\Box\neg j),$$

and that the T_1 condition (every singleton set is closed) is definable by

$$i \leftrightarrow \Diamond i.$$

Gabelaia [54] proves an analog of the Goldblatt-Thomason Theorem for the basic modal language with respect to topological semantics, and Sustretov [121] has extended the result to the basic hybrid language enriched with the universal modality. However Sustretov also shows that the T_2 condition (every distinct pair of points is contained in disjoint open neighbourhoods) is not definable in this richer language.

8 MODAL LOGIC AND ITS CHANGING ENVIRONMENT

Traditional motivations for and applications of modal logic came from philosophy, and dealt with such topics as modality, knowledge, conditionals, and obligations. Other strands dealt with more mathematical topics, leading to modal logics of time, space, or provability. As time went by, additional influences made modal logic even more diverse. Sources included computer science (for modal logics of computation and general processes), Artificial Intelligence (for modal logics for knowledge representation, non-monotonic reasoning, and belief revision), linguistics (for modal logics of grammatical structure), and the internet (for modal logics of trees). This web of new interfaces is still growing. Modern computer science, with its emphasis on new information carriers and networks of intelligent computing agents, also brings in modal logics of image processing, agency and security. And the empirical social sciences are joining in too, witness current applications of modal logic in economic game theory, or for modeling the powers of agents in social choice theory.

In the face of this diversity, the resilience of relational semantics is quite remarkable. Although nearly half a century old, its central ideas remain applicable, and applicable even when we enrich our conception of what a modal logic actually is. But what *are* the central ideas of relational semantics? In essence, this chapter has tried to make the following point clear: during the 50 or so years that relational semantics has existed, our understanding of it has become both broader and deeper. Originally conceived as a way of distinguishing and characterising logics (via soundness and completeness theorems) modal logicians have gradually unearthed the deeper mathematical themes that lie behind the seemingly modest facade of relational semantics; themes such as expressivity at the level of models versus the level of frames, the importance of bisimulation and other game-like constructions, the systematic links between the modal universe and many varieties of classical logic, ranging from first-order logic, through second-order logic, to the farther reaches of infinitary logic. Turning this perceived semantic unity into theorems is not always easy; work on combined modal logic still tends to be heavy on negative results, and first-order modal logic remains difficult territory. But unifying themes, such as guarding, and the possibility of applying ideas from abstract model theory, have emerged.

Indeed, we are tempted to conclude by playing devil's advocate: even the alternative semantics we have encountered indicate that something semantically central lies at the heart of relational semantics. For example, the Jónsson-Tarski Theorem reveals that relational semantics has an important algebraic core, and our excursion to the land of topological semantics revealed the centrality of the concept of bisimulation. Prediction is always a dangerous game (especially when it is about the future) but we believe that the interplay between theory and practice that has characterised research on modal logic throughout its history will continue to deepen our understanding of its semantic core. And, forced to place our bets, we would probably say: modal logics of games (see Chapter 20 of this handbook) will be a deep source of further insight, as will the coalgebraic semantics (discussed in Chapter 6).

ACKNOWLEDGEMENTS

We are extremely grateful to Carlos Areces for his painstaking editorial advice, to Aleksander Chagrov for his detailed list of corrections and recommendations, to Eric Kow for his patient work on the diagrams and formatting, to Sara Hobin for her meticulous proofreading and assistance with the cross-referencing, and to Dmitry Sustretov for checking Section 7. Special thanks to Daniel Gorín, Sergio Mera, Diego Figueira, and Santiago Figueira (the Buenos Aires crew) for giving the chapter its final trial.

BIBLIOGRAPHY

[1] M. Aiello and J. van Benthem. Logical patterns in space. In *Words, Proofs and Diagrams*, pages 5–25. CSLI Publications, 2002.

[2] M. Aiello, J. van Benthem, and G. Bezhanishvili. Reasoning about space: The modal way. *Journal of Logic and Computation*, 13(6):889–929, 2003.

[3] S. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, 1978.

[4] H. Andréka, I. Hodkinson, and I. Németi. Finite algebras of relations are representable on finite sets. *Journal of Symbolic Logic*, 64:243–267, 1999.

[5] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[6] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[7] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logics*, 123(1–3):287–299, 2003.

[8] A. Baltag. STS: a structural theory of sets. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*. CSLI Publications, 2000.

[9] J. Barwise and S. Feferman. *Model-Theoretic Logics*. Springer, 1985.

[10] J. Barwise and L. Moss. *Vicious Circles*. CSLI Publications, 1996.

[11] J. Barwise and J. van Benthem. Interpolation, preservation, and pebble games. *Journal of Symbolic Logic*, 29:881–903, 1999.

[12] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66(72), 1966.

[13] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[14] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fernmüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002, Copenhagen, Denmark, July/August 2002, Proceedings*, volume 2381 of *LNAI*, pages 38–52. Springer, 2002.

[15] P. Blackburn and M. Marx. Constructive interpolation in hybrid logic. *Journal of Symbolic Logic*, 68(2):463–480, 2003.

[16] W. Blok. The lattice of modal algebras: An algebraic investigation. *Journal of Symbolic Logic*, 45:221–236, 1980.

[17] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Compilers*, C-35(8), 1986.

[18] J. Burch, J. Clarke, K. McMillan, D. Dill, and J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *IEEE Symposium on Logic in Computer Science*, 1990.

[19] J. Burgess. Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic*, 22:76–84, 1979.

[20] S. Buss. Bounded arithmetic and propositional proof complexity. In H. Schwichtenberg, editor, *Logic and Computation*, pages 67–122. Springer, 1997.

[21] L. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, 56:1261–1272, 1991.

[22] A. Chandra and P. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proceedings 9th ACM STOC*, pages 77–90, 1977.

[23] C. Chang and H. Keisler. *Model Theory*. North-Holland Publishing Company, Amsterdam, 1973.

[24] B. Chellas. *Modal Logic, an Introduction*. Cambridge University Press, 1980.

[25] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[26] G. Corsi and S. Ghilardi. Semantical aspects of quantified modal logic. In *Knowledge, Belief, and Strategic Action*, pages 167–195. Cambridge University Press, 1992.

[27] M. Cresswell. Incompleteness and the Barcan formula. *Journal of Philosophical Logic*, 24:379–403, 1995.

[28] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In L. Aiello, J. Doyle, and S. Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 316–327. Morgan Kaufmann, San Francisco, California, 1996.

[29] M. de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57(2):566–584, 1992.

[30] M. de Rijke. *Extending Modal Logic*. PhD thesis, ILLC, University of Amsterdam, 1993.

[31] M. de Rijke. A Lindström theorem for modal logic. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra: A Bisimulation Perspective*, volume 53 of *Lecture Notes*, pages 217–230. CSLI Publications, 1995.

[32] E. Dijkstra. *A Discipline of Programming*. Prentice Hall, 1976.

[33] A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. *Theoretical Computer Science*, 311(1-3):221–256, 2004.

[34] M. Dummett and E. Lemmon. Modal logics between S4 and S5. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 5:250–264, 1959.

[35] H. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 1995.

[36] E. Emerson and R. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29:132–158, 1999.

[37] H. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York and London, second edition, 2001.
[38] L. Esakia. Diagonal constructions, Löb's formula, and Cantor's scattered spaces. In *Studies in Logic and Semantics*, pages 128–143. Metsniereba, 1981. In Russian.
[39] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.
[40] S. Feferman. Persistent and invariant formulas for outer extensions. *Compositio Mathematica*, 20:29–52, 1968.
[41] S. Feferman and Kreisel G. Persistent and invariant formulas relative to theories of higher order. *Bulletin of the American Mathematical Society*, 72:480–485, 1966.
[42] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36:331–346, 1970.
[43] K. Fine. An incomplete logic containing $S4$. *Theoria*, 40:23–29, 1974.
[44] K. Fine. Some connections between elementary and modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973.* North-Holland Publishing Company, 1975.
[45] K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 42(2):201–206, 1979.
[46] K. Fine and G. Schurz. Transfer theorems for stratified modal logics. In J. Copeland, editor, *Logic and Reality, Essays in Pure and Logic. In memory of Arthur Prior*, pages 169–213. Oxford University Press, 1996.
[47] M. Fischer and R. Ladner. Propositional modal logic of programs. In *Proceedings 9th ACM Symposium on Theory of Computation*, pages 286–294, 1977.
[48] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.
[49] M. Fitting and R. Mendelsohn. *First-Order Modal Logic*. Kluwer Academic Publishers, 1998.
[50] N. Friedman and J. Halpern. On the complexity of conditional logic. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation (KR '94)*, pages 202–213, 1994.
[51] D. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, 1981.
[52] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.
[53] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proceedings 7th ACM Symposium on Principles of Programming Languages*, pages 163–173, 1980.
[54] D. Gabelaia. Modal definability in topology. Master's thesis, University of Amsterdam, 2001.
[55] P. Gärdenfors and H. Rott. Belief revision. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, number 4, pages 35–132. Clarendon Press, 1995.
[56] S. Ghilardi. Incompleteness results in Kripke semantics. *Journal of Symbolic Logic*, 56:517–538, 1991.
[57] R. Goldblatt. Metamathematics of modal logic I. *Reports on Mathematical Logic*, 6:41–78, 1976.
[58] R. Goldblatt. Metamathematics of modal logic II. *Reports on Mathematical Logic*, 7:21–52, 1976.
[59] R. Goldblatt. Mathematical modal logic: a view of its evolution. *Journal of Applied Logic*, 1:309–392, 2003.
[60] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In J. Crossley, editor, *Algebra and Logic*, pages 163–173. Springer, 1974.
[61] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.
[62] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.
[63] E. Grädel and I. Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, pages 45–54, 1999.
[64] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time, I: lower bounds. *Journal of Computer and Systems Sciences*, 38(1):195–237, 1989.
[65] E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37:174–203, 1996.
[66] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.
[67] C. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:567–580, 1969.
[68] I. Hodkinson. Loosely guarded fragment of first-order logic has the finite model property. *Studia Logica*, 70:205–240, 2002.
[69] I. Hodkinson and M. Otto. Finite conformal hypergraph covers and Gaifman cliques in finite structures. *Bulletin of Symbolic Logic*, 9:387–405, 2003.
[70] E. Hoogland and M. Marx. Interpolation and definability in guarded fragments. *Studia Logica*, 70:373–409, 2002.
[71] E. Hoogland, M. Marx, and M. Otto. Beth definability for the guarded fragment. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, 6th International Conference LPAR99, Tbilisi, Georgia*, volume 1705 of *LNAI*, pages 273–285. Springer, 1999.
[72] M. Huth and M. Ryan. *Logic in Computer Science*. Cambridge University Press, 2nd edition, 1994.
[73] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to second order logic. In *Proceedings CONCUR '96*, volume 1119 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 1996.
[74] B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1952.
[75] B. Jónsson and A. Tarski. Boolean algebras with operators, Part II. *American Journal of Mathematics*, 74:127–162, 1952.

[76] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.
[77] S. Kanger. The morning star paradox. *Theoria*, pages 1–11, 1957.
[78] S. Kanger. *Provability in Logic*. Almqvist & Wiksell, 1957.
[79] B. Knaster. Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématiques*, 6:133–134, 1928.
[80] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
[81] D. Kozen and R. Parikh. A decision procedure for the propositional mu-calculus. In *Proceedings of the 2nd Workshop on Logic of Programs*, volume 164 of *LNCS*, pages 313–325. Springer, 1983.
[82] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal Logics. *Journal of Symbolic Logic*, 56:1469–1485, 1991.
[83] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.
[84] S. Kripke. Semantic analysis of modal logic I, normal propositional calculi. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
[85] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.
[86] S. Kripke. Naming and necessity. In *Semantics of Natural Language*. Reidel, 1972.
[87] N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.
[88] C. Lee. Representation of switching circuits by binary-decision programs. *Bell System Technical Journal*, 38:985–999, 1959.
[89] A. Lévy. *A Hierarchy of Formulas in Set Theory*, volume 57 of *Memoirs of the American Mathematical Society*. 1965.
[90] D. Lewis. *Counterfactuals*. Blackwell, 1973.
[91] P. Lindström. On extensions of elementary logic. *Theoria*, 35:1–11, 1969.
[92] D. Makinson. Some embedding theorems for modal logic. *Notre Dame Journal of Formal Logic*, pages 252–254, 1971.
[93] M. Marx. Tolerance logic. *Journal of Logic, Language and Information*, 6:353–373, 2001.
[94] M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, 1997.
[95] J. McKinsey and A. Tarski. The algebra of topology. *Annals of Mathematics*, 45:141–191, 1944.
[96] R. Montague. Pragmatics. In R. Klibansky, editor, *Contemporary Philosophy: a Survey*, pages 102–122. Florence, La Nuova Italia Editrice, 1968.
[97] R. Montague. Universal grammar. *Theoria*, 36:373–398, 1970.
[98] R. Moore. A formal theory of knowledge and action. In J. Hobbs and R. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood, New Jersey, 1985.
[99] M. Otto. Elementary proof of the van Benthem-Rosen characterisation theorem. Technical Report 2342, Fachbereich Mathematik, Technische Universität Darmstadt, 2004.
[100] D. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, pages 167–183. Springer, 1981.
[101] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.
[102] V. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings 17th IEEE Symposium on Computer Science*, pages 109–121, 1976.
[103] V. Pratt. Models of program logics. In *Proceedings 20th IEEE Symposium on Foundations of Computer Science*, pages 115–222, 1979.
[104] A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.
[105] A. Prior. *Papers on Time and Tense*. Oxford University Press, New edition, 2003. Edited by Hasle, Øhrstrom, Braüner, and Copeland.
[106] W. Quine. Three grades of modal involvement. In *The Ways of Paradox and Other Essays*, pages 156–174. Random House, 1953.
[107] M. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.
[108] G. Restall. *An Introduction to Substructural Logics*. Routledge, 2000.
[109] E. Rosen. Modal logic over finite structures. *Journal of Logic, Language and Information*, 6:427–439, 1997.
[110] M. Ryan and P. Schobbens. Counterfactuals and updates as inverse modalities. *Journal of Logic, Language and Information*, 6:123–146, 1997.
[111] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973.*, pages 110–143. North-Holland Publishing Company, 1975.
[112] D. Scott. Advice on modal logic. In K. Lambert, editor, *Philosophical Problems in Logic*, pages 143–173. Reidel, 1970.
[113] K. Segerberg. *An Essay in Classical Modal Logic*. Filosofiska Studier 13. University of Uppsala, 1971.
[114] K. Segerberg. Two-dimensional modal logics. *Journal of Philosophical Logic*, 2:77–96, 1973.
[115] V. Shehtman. Modal logics of domains of the real plane. *Studia Logica*, 42:63–80, 1983.

[116] V. Shehtman. “Everywhere” and “Here”. *Journal of Applied Non-classical Logics*, 9:369–380, 1999.

[117] V. Shehtman and D. Skvortsov. Semantics of non-classical first-order predicate logics. In *Mathematical Logic*, pages 105–116. Plenum Press, New York, 1990.

[118] E. Spaan. *Complexity of Modal Logics*. PhD thesis, ILLC, University of Amsterdam, 1993.

[119] R. Street and E. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81:249–2644, 1989.

[120] H. Sturm. The true bisimulations for 'since' and 'until'. *Logic and Logical Philosophy*, 10:173–183, 2002.

[121] D. Sustretov. Hybrid definability in topological spaces. Technical report, University of Amsterdam, 2005. ILLC Technical Report PP-2005-14.

[122] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[123] Alfred Tarski. Der Aussagenkalkül und die Topologie. *Fund. Math.*, 31:103–134, 1938.

[124] B. ten Cate. *Model Theory for Extended Modal Languages*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2004.

[125] S. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40:150–158, 1974.

[126] S. Thomason. Reduction of second-order logic to modal logic. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 21:107–114, 1975.

[127] A. Urquhart. Decidability and the finite model property. *Journal of Philosophical Logic*, 10:367–370, 1981.

[128] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[129] J. van Benthem. Two simple incomplete modal logics. *Theoria*, 44:25–37, 1978.

[130] J. van Benthem. Canonical modal logics and ultrafilter extensions. *Journal of Symbolic Logic*, 44:1–8, 1980.

[131] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.

[132] J. van Benthem. Beyond accessibility: Functional models for modal logic. In *Diamonds and Defaults*, pages 1–18. Kluwer, 1993.

[133] J. van Benthem. Modal frame classes revisited. *Fundamenta Informaticae*, 18:307–317, 1993.

[134] J. van Benthem. Bisimulation: the never ending story. In J. Tromp, editor, *A dynamic and quick intellect. Liber Amicorum Paul Vitányi*, pages 23–27. CWI Amsterdam, 1996.

[135] J. van Benthem. Dynamic bits and pieces. Technical Report LP-97-01, Institute for Logic, Language and Computation, University of Amsterdam, 1997.

[136] J. van Benthem. Modal foundations for predicate logic. *Bulletin of the IGPL*, 5(2):259–286, 1997.

[137] J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, number 3, pages 325–408. Kluwer Academic Publishers, 2001.

[138] J. van Benthem. Minimal predicates, fixed points and definability. *Journal of Symbolic Logic*, 70:696–712, 2005.

[139] J. van Benthem. A new modal Lindström theorem. Technical Report PP-2006-06, Institute for Logic, Language and Computation, University of Amsterdam, 2006.

[140] M. Vardi. On the complexity of epistemic reasoning. In *Proceedings 4th IEEE Symposium on Logic in Computer Science*, pages 243–252, 1989.

[141] M. Vardi. On the complexity of bounded variable queries. In *Proceedings 14th ACM Symposium on Principles of Database Systems*, pages 266–276. 1995.

[142] M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31*, pages 149–184. AMS, 1997.

[143] F. Veltman. *Logics for Conditionals*. PhD thesis, University of Amsterdam, 1985.

[144] F. Wolter. Fusions of modal logics revisited. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, pages 361–369. CSLI Publications, 1998.

[145] G. von Wright. *An Essay in Modal Logic*. North-Holland Publishing Company, 1951.

INDEX