Modal Logic: A Semantic Perspective

Patrick Blackburn and Johan van Benthem

Abstract

This chapter introduces modal logic as a tool for talking about graphs, or to use more traditional terminology, as a tool for talking about Kripke models and frames. We want the reader to gain an intuitive appreciation of this perspective, and a firm grasp of the key technical ideas (such as bisimulations) which underly it. We introduce the syntax and semantics of basic modal logic, discuss its expressivity at the level of models, examine its computational properties, and then consider what it can say at the level of frames. We then move beyond the basic modal language, examine the kinds of expressivity offered by a number of richer modal logics, and try to pin down what it is that makes them all 'modal'. We conclude by discussing an example which brings many of the ideas we discuss into play: games.

Contents

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. Moreover, modal logic can be given a *topological semantics*, so it can also be viewed as a tool talking about certain kinds of topologies. But this chapter is about 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 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 states, 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 a 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 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 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 main 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 think 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 locate the key ideas in a wider logical context, and even beginners should try to follow them.

We proceed as follows. 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 such 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 bisimulations 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 links between modal and classical logic from a number of different perspectives. We learn that there is an important connection between modal logic and (monadic) second-order logic, discuss correspondences between modal logic and first-order logic with fixed-point operators, and show that first-order logic itself can be viewed as a modal 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 themes explored in earlier sections re-emerge, and point towards an idea that seems to lie at the heart of modal logic: *guarding*. Section 7 closes the chapter with a brief discussion of the changing role of modal logic.

One final remark. We *don't* discuss modal proof-theory or related notions such as completeness in any detail (these topics are the focus of Chapter **??** 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 I 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

Given proposition symbols PROP $= \{p, q, r, \ldots\}$, and modality symbols MOD $= \{m, m', m'', \ldots\}$ (the choice of PROP and MOD is often called the *signature* or *similarity type*) we define the *basic modal language* (over this signature) as follows:

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

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 $[m]\varphi$ is equivalent to $\neg\langle m \rangle \neg\varphi$. 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".

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$, where W is a non-empty set (whose elements we usually call *points*), each R^m 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, and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, R, V)$. 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):

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

A formula φ is *globally 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 true 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 true, 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.

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 quantification 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' intuition precise.

Third, note that Kripke models can (and in our opinion should) be thought of as graphs. As we have already mentioned, modal logic has been applied in many different area. 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 points in W as times, and the relation R as the relation of temporal precedence (that is, Rww' means that the time w is earlier than time w'). Consider the graph in Figure 1. This shows a simple flow of time consisting of five points. Here we will

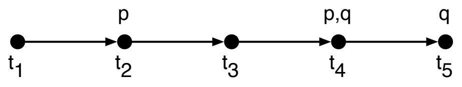

Fig. 1. A simple temporal model

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 formulas $\Diamond(p \wedge q)$ 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 [39]. His work offers what is probably the clearest example of modal logic being appreciated for the 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.

Our next example brings us to one of the currently most influential ways of thinking about Kripke models; to view them as pictures of computational systems (we examine to this perspective in more detail in Section 6 when we discuss Propositional Dynamic Logic and the modal μ-calculus, and the idea underlies Chapter **??** of this handbook). Consider the graph shown in Figure 2. This shows a finite state automaton for the language

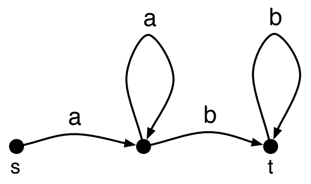

Fig. 2. Finite state automaton for $a^n b^m (n, m > 0)$

$a^n b^n$ $(n, m > 0)$, 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. It 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

$$\langle a \rangle \cdots \langle a \rangle \langle b \rangle \cdots \langle b \rangle \top$$

(that is, an unbroken block of $\langle a \rangle$ diamonds preceding an unbroken block of $\langle b \rangle$ diamonds) are satisfied at the start node s as all (and indeed only) 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 non-deterministic models of processes, and we shall do so in Section 6.

Another important application of modal languages is to model the logic of knowledge and belief. Again, simple graph-based intuitions underly this application. Consider, for example, the graph shown in Figure 3. Here we see the epistemic states of a very simple agent. One state, the agent's current state, is marked e. This represents the agents current knowledge (the agent knows that q is the case). The other states represent the way the world might be. For example, although neither p nor r are true in the current state, the agent views states in which p and q, and r and q (but not p and q and r together) as epistemically acceptable alternatives to the current state. That is $\Diamond(p \wedge q)$ ("$p \wedge q$ is consistent with what the agent knows") and $\Diamond(r \wedge q)$ are both satisfied at e. Moreover $\Box p$ ("the agent knows that p") is satisfied at e, as at every epistemic alternative the information p holds.

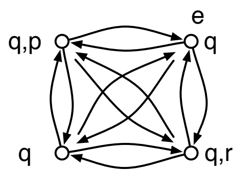

Fig. 3. Epistemic states of a simple agent

The next example is important for another reason. Modal logic is often veiwed 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 is "necessarily", and the points of the Kripke model are regarded as possible worlds. Unfortunately, this interpretation has tended to overshadow the others, at least in certain research communities (some philosophers seem to view modal logic, intensionality, 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.

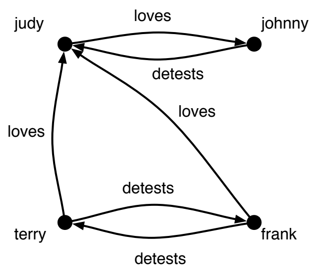

Fig. 4. Ordinary individuals

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

$$\langle \textsc{loves} \rangle \top \wedge \langle \textsc{detests} \rangle \langle \textsc{loves} \rangle \top$$

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

We're almost ready to define the standard translation, but before doing so let's deal with two other matters. First, in most branches 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 2.1 [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 letters 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 2.2 *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 3.5, for an example of such a proof.) $\dashv$

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.

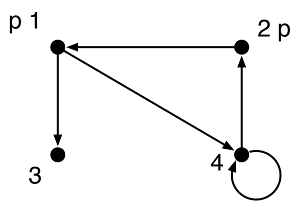

Fig. 5. The formula $\Diamond\Box\Diamond p$ is true at 1 and 4, but false at 2 and 3

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 player 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 force F to make both disjuncts false; negation switches the roles of the two players; and diamonds makes V pick a successor of the current point, while boxes do the same for F. Moreover, for any propositional 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.

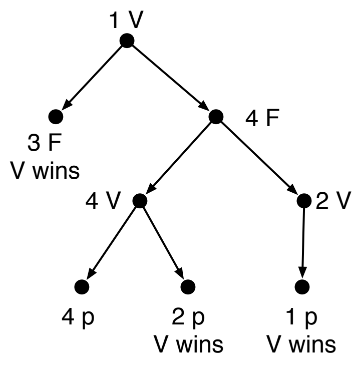

Fig. 6. Initial segment of a game tree

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 suggest is completely general: for any modal $\mathfrak{M}$, 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.

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 this 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. We might then point out that the set of all modal validities (that is, the *minimal modal logic*) can be axiomatised by a Hilbert-style proof system called **K**. The axioms of **K** are:

(i) All propositional tautologies,

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

There are two rules of proof: *modus ponens* (if $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$) and *modal generalisation* (if $\vdash \varphi$ then $\Box\varphi$). 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 and terms, such as $\forall x\varphi \to [t/x]\varphi$. This is no coincidence. As the standard translation 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 all $p \in$ 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 $(W, \{R^m\}_{m\in\text{MOD}})$ 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) can corresponds to a first-order formula from this language via the *standard translation*:

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

That is, the standard translation maps propositional 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 simply expands these macros. Note that $\text{ST}_x(\varphi)$ 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 modal at a certain point.

Here's an example of the translation at work:

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

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

Proposition 2.3 *For any 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. $\dashv$

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

Proposition 2.4 *Basic model 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. Suppose the Löwenheim-Skolem property fails. Then there is a set of basic modal formulas Σ that has at least one infinite model, but lacks models of some infinite cardinalities. But this is impossible. 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 2.3 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 2.3, each of these models satisfies Σ, and we conclude that basic modal logic must have the Löwenheim-Skolem property after all. The argument showing that it has the Compactness property is analogous. $\dashv$

Another easy consequence of the standard translation is that the set of validities (in basic modal languages) is recursively enumerable. Again, by appealing to the fact that first-order logic has this property, we swiftly generate a contradiction from the supposition that basic modal languages do not.

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 access 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 model 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 Simulation 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 (in these graphs all points are irreflexive).

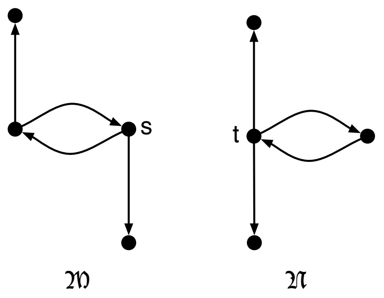

Fig. 7. $\mathfrak{M}$ and $\mathfrak{N}$ are modally distinguishable.

Here $\Box(\Box\perp\vee\Diamond\Box\perp)$ 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 (in these graphs, u is reflexive, and all other points are irreflexive).

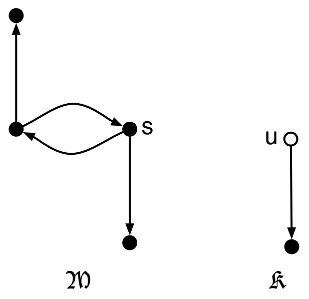

Fig. 8. $\mathfrak{M}$ and $\mathfrak{K}$ are not modally distinguishable.

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 is easy to distinguish them if we are 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, 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 *Structural invariances: 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 model $\mathfrak{M}$ and $\mathfrak{K}$ of Figure 7 provide an intuitive example: they seem to stand for the same process when we look at possible actions and deadlocks. At each live stage, the process can opt for a deadlock. By contrast, $\mathfrak{M}$ and $\mathfrak{N}$ are different, as not every state has an immediate dead-lock option. Or consider the epistemic interpretation: when would we want to say that two graphs represent the same epistemic information? For example we would probably want to identify the two epistemic models shown in Figure 9 at their distinguished points s and t.

After all, in essence both models present us with a two way choice: either we are in a p knowledge state, and there is a distinct knowledge state q that is compatible with what we know, or vice versa. The intuition that both these diagrams code the same information is captured by our modal language: the reader will not find any modal formula that distinguishes them.

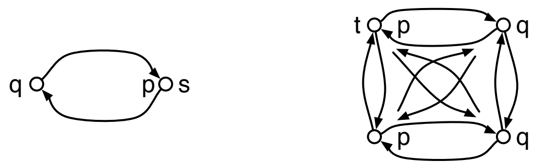

Fig. 9. Two epistemically equivalent models.

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, which is best understood as a notion of equivalence between modal models viewed as process graphs. We state it here for models with one relation R, but the definition generalises straightforwardly to models with any number of atomic relations.

Definition 3.1 (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 points (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.

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 at 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 3.1 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 define a quotient of $\mathfrak{M}$ whose points are the equivalence classes of the maximal bisimulation on $\mathfrak{M}$ itself, setting $|w|R|v|$ iff the equivalence classes $|w|$ and $|v|$ contain points w' and v' such that $w'Rv'$. 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, 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.)

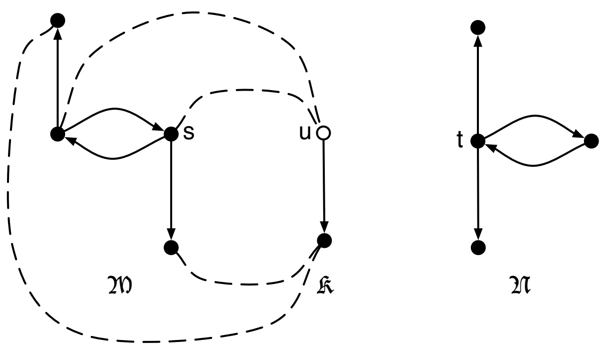

Fig. 10. $\mathfrak{M}$ and $\mathfrak{K}$ are bisimilar, $\mathfrak{K}$ and $\mathfrak{N}$ are not.

But bisimulations can also be used to make bigger models: a pointed model $(\mathfrak{M}, w)$ can be made as large as possible using a construction called *tree unraveling*. To unravel a model we 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 $\mathfrak{M}$. As an example, consider the unraveling of 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 unique successor of u).

$$\begin{tikzcd}
<u> \arrow[r] \arrow[d] & <u,v> \\
<u,u> \arrow[r] \arrow[d] & <u,u,v> \\
<u,u,u> \arrow[r] & <u,u,u,v> \\
\vdots &
\end{tikzcd}$$

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

Reasoning about trees is often easier than reasoning about arbitrary graphs, and so this method is of considerable theoretical utility. Moreover, as we shall see, 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 decade before the unifying concept of a bisimulations was introduced. All three constructions are fundamental tools in many areas of modal logic (for example, 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, and form the disjoint union of the copies (after all, in modal logic we are only interested in models up to isomorphism). This lumping together process can be generalised to arbitrarily many models, which prompts the following definition.

Definition 3.2 (Disjoint Unions) *Given a collection of 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 identify 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.

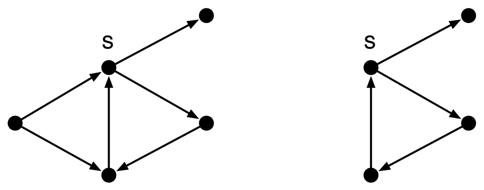

Fig. 12. A generated submodel.

It is clear that 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 3.3 (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 symbol 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 3.4 (Bounded morphism) *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 $R'f(u)f(v)$.*

Zag: *if $R'w'v'$, 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 a 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).

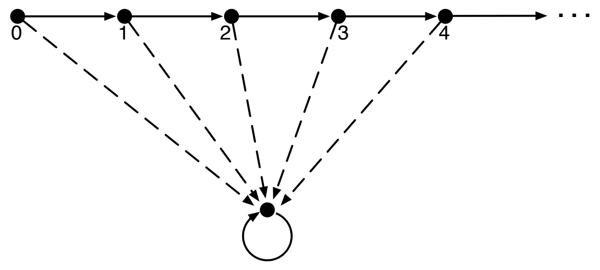

Fig. 13. Bounded morphism collapsing the natural numbers to a reflexive point.

Here we have collapsed the natural numbers in their usual order to a single reflexive point. It 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 2.2) modal formulas are invariant for isomorphism. Moreover, 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:

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

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 elementary 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 its formulas 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 and a

Duplicator. 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 forgo the 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 3.5 (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, but uses zag in place of zig. $\dashv$

The result allows us to quickly show failures of bisimulation. For example, we have already sketched an argument showing that the models $\mathfrak{M}$ and $\mathfrak{N}$ of Figure 7 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 $\Box(\Box\perp \vee \Diamond\Box\perp)$) which is 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 $\mathfrak{K}$.

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

Lemma 3.6 *Modal satisfaction is invariant under the formation of disjoint unions, generated submodels, and bounded morphisms. That is:*

(i) *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 $\mathfrak{M}_i, w \models \varphi$.*

(ii) *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 $\mathfrak{M}', w' \models \varphi$.*

(iii) *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 3.5. $\dashv$

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

Proposition 3.7 *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 shall construct the required bisimulation by showing that the relation of modal equivalence is itself a bisimulation. That is, we

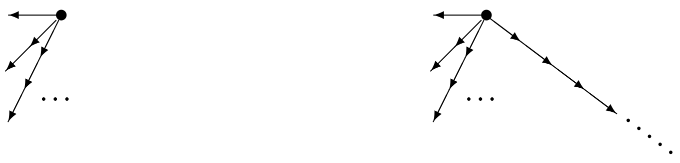

Fig. 14. Modally equivalent but not bisimilar.

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 $\{w'_1, \ldots, w'_n\}$. By assumption, for every $w'_i \in S'$ there exists a formula ψ_i such that $\mathfrak{M}, v \models \psi_i$ but $\mathfrak{M}', w'_i \not\models \psi_i$. 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.

$\dashv$

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 finally many R-successors) and suitably saturated models in a model-theoretic sense. But no general converse can hold, for the reason mentioned earlier for first-order logic. Indeed, the converse does not hold generally even for countable models: not all modally equivalent countable models are bisimilar. The two models in Figure 3.4 satisfy the same modal formulas at their roots, but if there were a bisimulation between them, the infinite chain on the right would also have to occur on the left.

This counterexample can be repaired by passing to an *infinitary model language* $\mathcal{L}_\infty$ 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 $\mathcal{L}_\infty$ 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 example are *propositional dynamic logic* and *modal μ-calculus*, which are discussed in Section 6.

Lemma 3.5 and its partial converses do not exhaust 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. Let's do this right away,

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 $ST_x(\varphi)$ 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 can be viewed as a proper expressive fragment of the corresponding first-order language, exactly which fragment is it? This question has an elegant answer. First, a preliminary definition.

Definition 3.8 *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 $\mathfrak{M} \models \varphi(x)[w]$ iff $\mathfrak{M}' \models \varphi(x)[w']$.*

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

Theorem 3.9 (Modal Characterisation Theorem) *The following are equivalent for all first-order formulas $\varphi(x)$ in one free variable x:*

(i) *$\varphi(x)$ is invariant for bisimulation.*

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

Proof. That clause *(ii)* implies *(i)* is a more or less immediate consequence of Lemma 3.5. The hard direction is showing that clause *(i)* implies *(ii)*. A model-theoretic proof of this result is given in Chapter **??** of this handbook. $\dashv$

Nowadays many different proofs are known for this result, and for various extensions and variants. In particular, the result also holds in finite model theory where the standard model-theoretic results (such as compactness) cannot be applied. More recently it was shown that the modal equivalent in the clause *(ii)* can be restricted to a formula of modal operator depth 2^K, where k is the quantifier depth of $\varphi(x)$. Incidentally, determining whether a given first-order formula is equivalent to a modal one is *undecidable*. This complexity is not as bad as it sounds, because the same is true for most significant fragments of first-order logic that are defined semantically.

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 properties of its more powerful neighbour, such as the Compactness and Löwenheim-Skolem theorems. But not all such transfer is automatic. Consider 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 $\theta \models \psi$.

If we want the same result for modal formulas such that $\varphi \models \psi$, this result gives us a first-order formula θ such that $ST_x(\varphi) \models \theta$ and $\theta \models ST_x\psi$. But what guarantees that this interpolant is modally definable? Interpolation does in fact hold for the basic modal language, but additional work is needed to prove this.

While we are on the topic, it's worth mentioning that interpolation meshes well with the above preservation results. Here is an improvement on the Modal Invariance Theorem. Let us say that a first-order formula φ *implies ψ along bisimulation* if the following implication always holds: if E is a bisimulation between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$, and $\mathfrak{M}, s \models \varphi$, then $\mathfrak{N}, t \models \varphi$.

Theorem 3.10 (Modal Preservation/Interpolation Theorem) *The following are equivalent for all first-order formulas $\varphi(x)$:*

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

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

The Modal Invariance Theorem follows by taking $\varphi(x)$ equal to $\varphi(x)$. This result does not imply ordinary modal interpolation as it stands: additional work is needed again.

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 proof has been found so far significant formulations of this idea, 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.

We conclude this discussion with two general transfer 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 3.11 (First Lifting Lemma) *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

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

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

Lemma 3.12 (Second Lifting Lemma) *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

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

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

The first result is the key item in (some proofs of) the Characterisation Theorem (the $^+$-models are suitably saturated elementary extensions). The second result 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

We have said that bisimulation is a sort of process equivalence. The dynamic character of the notion can be brought out by viewing it as a game. Consider a game between Duplicator (the analogy player) and Spoiler (the difference player) comparing successive pairs (s, t) in two model $\mathfrak{M}$ and $\mathfrak{N}$:

> *In each round Spoiler chooses a state u in one model which is a successor of the current s or t, and Duplicator responds with a matching successor v in the other model. If u and v differ in their atomic properties, Spoiler wins. If Duplicator cannot find a matching successor, Spoiler wins as well.*

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, retuning 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:

Theorem 3.13 (Adequacy of modal comparison games)

(i) *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.*

(ii) *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 $\mathfrak{N}$ linking s to t.*

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 $\Box(\Box \perp \vee \Diamond\Box \perp)$ of modal operator depth three. In each round he can make sure that some modal difference remains at the current match, with the 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; 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: the positive computational results reported here arise from negative expressivity results, such as the inability of the basic modal language to force the existence of infinite models.

4.1 Model checking

Here's a simple formulation of model checking task (for the basic modal language):

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?

But is this 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 atomic 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 tests whether the abstract information embodied in the formula is implicitly present in the model. 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 salutory 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 directly expressed in terms of model checking. A classic example is hardware verification. A chip can be thought of as a model: abstractly conceived, a chip is a (very large) collection of entities, bearing certain properties, and related in various ways. But if a chip is to work satisfactorily, it should also possess a number of high-level 'emergent' properties: for example, it should not deadlock. If we have a modal language that can express the desired properties, then by checking the formula in the model we can determine whether the chip is well-designed.

So how should we perform model checking? There's a good way and a bad way — let's start with the bad. If we are computationally naive, we might proceed as follows: we'd write a program that takes a model $\mathfrak{M}$, a point w, and a formula φ, and then applies the satisfaction definition in a top-down fashion to φ and all its subformulas. That is, the program would recursively break φ down into its components and evaluate them in the manner described in the satisfaction definition; in particular, each occurrence of a $\Box$ or $\Diamond$ prefixed subformula would be evaluated at the relevant accessible points.

To see why this approach can be bad, consider the following configuration of points:

Suppose we evaluate $\Box\Diamond\varphi$ at t. As we are working top down, this means we have to evaluate $\Diamond\varphi$ at both u and v. But this in turn means we will evaluate φ at w twice, once while working from u, and once while working from v. This (senseless) re-evaluation of formulas renders the naive top-down algorithm impractical: it is liable to be swamped by unnecessary work.

$$\begin{tikzcd}
& u & \\
w \arrow[ur] & & t \arrow[ul] \\
& v \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Fortunately, there's a better way. Instead of working top-down, use a bottom-up *labelling algorithm*: simply label every node in the model with all of the subformulas φ that are true there. 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. 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 $\mathfrak{M}$. Note, incidentally, that the algorithm actually supplies us with a more detailed answer than we asked for: instead of just telling us whether φ is true in $\mathfrak{M}$ at w, it gives us a complete listing of all the points (if any) where φ is true in $\mathfrak{M}$.

The labelling algorithm adapts, fairly straightforwardly, to many more powerful modal languages. As we said above, when modal checking we want to work with a language capable of expressing interesting high-level properties, and the ordinary $\Box$ and $\Diamond$ usually aren't strong enough. Far more useful is the binary *Until* modality:

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

(Here R^* is the reflexive transitive closure of an irreflexive accessibility relation R, and R^+ is its transitive closure.) The *Until* modality (which comes in several related forms) has proved useful in many model checking applications, and we can label for it as follows. First, if any point w is labeled with ψ, label w with $U(\psi, \theta)$ for all subformulas θ of the input formula. Second, if any point w is labeled with θ and at least one R-successor of w is labeled with $U(\psi, \theta)$, then label w with $U(\psi, \theta)$.

Throughout the above discussion (and indeed, in the discussion that follows) 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 this: 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 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 *Ordered Binary Decision Diagrams* (OBBDs), a compact representation for boolean expressions, together with sophisticated algorithms for working with them. The use of OBDDs in model checking lead to a breakthrough in the early 1990s in the size of the models that could be handled. The reader should not underestimate the gap between the labelling algorithm sketched above, and what it takes to make 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 exists that explores the issues involved.

But let's conclude our discussion on another note: what does model checking look like from a first-order perspective? That is, how difficult is it to model check if we are free to give an arbitrary first-order formula as input? A little thought reveals that the task is likely to be difficult: there seems to be no simple way to extend the labelling algorithm to handle the quantifiers. And in fact, model checking first-order formulas is a PSPACE-complete task. That is, it is possible to write an algorithm that solves the task using an amount of computer memory that is only polynomial in the size of the input model and formula, though the algorithm may well require time exponential in the size of the input. Incidentally, there does not seem to be much work on first-order model checking. First-order logic, although so much more complex, does not seem to offer the kind of expressivity required for typical model checking applications. The *Until* family of modalities offer the basic expressivity required, and when more is needed the natural next step is to move to the modal μ-calculus, a richer modal language which enables fixed points to be dealt with (something that is beyond the scope of first-order logic). Surprisingly, it was recently shown that the model checking task for modal μ-calculus can also be carried out polynomial time.

4.2 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 model 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:

$$ST_x(\Diamond\varphi) = \exists y\,(Rxy \wedge ST_y(\varphi)) \qquad ST_y(\Diamond\varphi) = \exists x\,(Ryx \wedge ST_x(\varphi))$$

$$ST_x(\Box\varphi) = \forall y\,(Rxy \rightarrow ST_y(\varphi)) \qquad ST_y(\Box\varphi) = \forall x\,(Ryx \rightarrow ST_x(\varphi)).$$

The interleaving of ST_x and ST_y guarantees that for any basic modal formula φ, $ST_x(\varphi)$ will contain only the two variables x and y, and it should be clear that the modified translation is equivalent to the original translation. 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 doesn't tell us very much about *why* modal logic is decidable. The following semantic argument is 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

$$\forall x \neg Rxx \wedge \forall x \forall y \forall z (Rxy \wedge Ryz \to Rxz) \wedge \forall x \exists y Rxy$$

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 4.1 (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 $2^{s(\varphi)}$ 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 4.2 (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 $W_\Sigma = \{|w|_\Sigma \mid w \in W\}$. Suppose $\mathfrak{M}^f_\Sigma$ is any model (W^f, R^f, V^f) such that:

(i) $W^f = W_\Sigma$.

(ii) *If Rwv then $R^f|w||v|$.*

(iii) *If $R^f|w||v|$ then for all $\Diamond\varphi \in \Sigma$, if $\mathfrak{M}, v \models \varphi$ then $\mathfrak{M}, w \models \Diamond\varphi$.*

(iv) *$V^f(p) = \{|w| \mid \mathfrak{M}, w \models p\}$, 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 $|\Sigma|$. 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, note that the previous definition does *not* specify an accessibility relation on W_Σ — it only imposes constraints (namely clauses (ii) and (iii)) on the properties a suitable accessibility relation R^f should have. That the constraints imposed are sensible is shown by the following result:

Theorem 4.3 (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 that Σ is closed under subformulas, the inductive step for the boolean connectives is immediate.

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 (ii) 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 (iii) of the previous definition) we conclude that $\mathfrak{M}, w \models \Diamond\sigma$. $\dashv$

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

(i) $R^s|w||v|$ iff $\exists w' \in |w| \exists v' \in |v|\, Rw'v'$.

(ii) $R^l|w||v|$ 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 4.1: the basic modal language indeed has the strong finite model property. As we argued above, this is turn shows 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: decidability is regained at the expense of expressivity.

4.3 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 nondeterministically choose a model containing at most $2^{s(\varphi)}$ points, and then check whether or not it satisfies φ (which takes time exponential in the size of φ). 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 not NEXPTIME-complete, it 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 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 the modal satisfiability problem is probably much 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 4.4 *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 of modal depth less than 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 4.5 *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 $\mathit{md}(\varphi)$ away from the root we obtain a finite model which (by the previous lemma) satisfies φ at its root. $\dashv$

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 $\mathit{md}(\varphi)$) 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 constructed, such an algorithm runs in PSPACE. 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.

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 some 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 emphasized by the description logic community. Indeed, description logic builds into its architecture the idea of a *T-Box*, a collection of (multi-modal) 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 T-Box 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 bretheren. To put it another way, suppose we have the following T-Box:

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

Let's call this T-Box 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

$$\langle \textsc{mother} \rangle (\mathit{bear} \wedge \mathit{human})?$$

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

$$\langle \textsc{fedby} \rangle (\neg human \wedge \neg \mathit{mother})?$$

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

Local-global satisfaction 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. 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 4.6 *Suppose we are working with a multi-modal language $\mathcal{L}$ in which all the modalities are unary and are interpreted by arbitrary binary relations. Then both the local-global satisfiability task for $\mathcal{L}$ and the global satisfiability task for $\mathcal{L}$ are EXPTIME-complete.*

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 measure. One of the most interesting recent developments in computational logic has come from the description logic community, who have shown it is possible to specify and implement algorithms for such problems that are remarkably efficient in practice.

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 testing, 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 special complexity class of its own.

Testing for bisimulation, however, turns out to be easy. There are elegant polynomial algorithms which work by discarding pairs of point that cannot make it into any bisimulation. 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.

5 Richer logics

Until now, we have been acting as if there was merely one modal logic, namely the set of formulas true in every model or (to put it syntactically) the set of formulas generated by the system **K**. But traditional presentations of modal logic tend to emphasise the *multiplicity* of modal logic. Nowadays most attention is devoted to logics richer than **K**, for example logics such as **T**, **K4**, **S4**, **S5**, **GL**, and **Grz**. Logics weaker than **K** are studied too, but we won't say anything about them here.

Where do these richer logics come from? Basically, from the level of *frames*. 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 times, it is reasonable to assume that the accessibility relation is transitive, and (as the reader can easily check) the formula $\Box p \rightarrow \Box\Box p$ cannot be falsified. As this formula is not provable in **K**, if we want a logic for working with transitive temporal flows we should add it as an extra axiom; doing so gives us the logic **K4**. We shall begin this section by discussing such axiomatic extensions of **K** in a little further.

But this chapter is about expressivity, not proof systems, and as we shall see there is a fundamental expressive distinction between the level of models and the level of frames: whereas modal logic at the level of models is essentially the (bisimulation invariant fragment of) first-order logic, modal logic at the level of frames is essentially a fragment of second-order logic.

5.1 Axioms and relational frame properties

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

Theorem 5.1 *A formula is provable in **K4** (that is, **K** plus all instances of the axiom schema $\Box\varphi \rightarrow \Box\Box\varphi$) iff it is true in all models based on frames whose accessibility relation is transitive.*

That is, the theorems of **K4** are true in all graphs with a transitive relation, while its non-theorems have some transitive counter-example; the additional axiom $\Box\varphi \rightarrow \Box\Box\varphi$ reflects a simple visualisable geometric condition 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, to various types of model surgery (such as filtration and unraveling). 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** plus the Löb axiom $\Box(\Box p \rightarrow p) \rightarrow p$, which arose via the study of arithmetical provability, and was later proved complete with respect to the class of finite trees). Sometimes, however, we might start with a natural model class — say an interesting space-time structure — and try axiomatise its modal validities. The literature is replete with both variants.

Nowadays a lot is known about axiomatic extensions of **K**. For start, it turns out that there are uncountably many such *normal modal logics*, as they are often called. The cartography of this landscape is an object of study in its own right; here we shall only mention that it contains two major highways, because of the following result due to Makinson:

Theorem 5.2 *Every normal modal logic is either a subset of the logic **Id** (with characteristic axiom $\varphi \leftrightarrow \Box\varphi$ or of **Un** (with characteristic axiom $\Box\perp$).*

The systems **T**, **K4**, **S4**, and **S5** lie on the first road, and **GL** lies on the second.

But perhaps the most interesting fact to have emerged about normal modal logics is that not all of them have frame-based characterisations. Frame completeness results (such as the theorem for **K4** noted above) are the exception, rather than the rule. We won't explore such *frame incompleteness* results further here, but the underlying source of them is the second order expressivity that modal logic exhibits at the level of frames, the topic to which we now turn.

5.2 Frame correspondence and second-order logic

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 the first-order theory of (say, linear order) is constructed over the set of first-order validities. Nothing of substance hangs on this, but it fits more naturally with our focus on expressivity. We say very little about deduction in what follows; we will simply investigate what modal formulas can say about frames.

First some terminology and notation. We shall call a modal formula $\varphi(p_1, \ldots, p_n)$ *true in a frame* $\mathfrak{F} = (W, R)$ *at a world* s if, for each valuation V for its proposition symbols $p_1,\ldots,p_n$, we have that φ holds in the model; in such a case we write $\mathfrak{F}, s \models \varphi$. We shall call a modal formula φ *true in a frame* $\mathfrak{F}$ (or *valid in* $\mathfrak{F}$) if it is true at each point in $\mathfrak{F}$, and we write this as $\mathfrak{F} \models \varphi$. Moreover, we say that a modal formula is *true* (or *valid*) on a class of frames F if it is true on each frame $\mathfrak{F}$ in F. Finally, we say that φ *defines* a class of frames if it is true on precisely the frames in $\mathfrak{F}$.

Let's consider some examples. $\Box p \rightarrow \Box\Box p$ defines the class of transitive frames (or more simply: defines transitivity). for a straightforward argument shows that

$$\mathfrak{F}, s \models \Box p \rightarrow \Box\Box p \text{ iff } \mathfrak{F} \models \forall y(Rxy \rightarrow \forall z(Ryz \rightarrow Rxz))[s].$$

Similarly, $\Box p \rightarrow p$, the **T** axiom, defines the class of reflexive frames (or: defines reflexivity) for an even easier argument shows that

$$\mathfrak{F}, s \models \Box p \to p \text{ iff } \mathfrak{F} \models \forall x(Rxy \to \forall z(Ryz \to Rxz))[s].$$

Furthermore, it is simple to see that $\varphi \leftrightarrow \Box\varphi$, the **Id** axiom, defines the class of frames consisting of a collection of isolated reflexive points, and that $\Box \perp$, the **Un** axiom, defines the class of frames consisting of a collection of isolated irreflexive points.

Note that all four classes of frames are definable by simple first-order formulas — and this is actually rather puzzling. After all, if we think about the definition given above of what it means for a formula $\varphi(p_1, \dots, p_n)$ to be true in a frame, we see that this concept is essentially *second-order*: we quantify across all valuations, and valuations assign *subsets* of frames to proposition symbols.

We can make this second-order perspective precise in terms of our Standard Translation ST_x over models: frame truth treats modal formulas φ as *monadic second-order* closures of their standard first-order translations on relational models, that is, as monadic Π^1_1 formulas of the form

$$\forall P_1 \cdots P_x ST_x \varphi.$$

Now, some well known axioms do involve genuine non-first-order conditions. A famous case is Löb's axiom, $\Box(\Box p \to p) \to \Box p$. This defines the conjunction of the transitivity of R with the upward well-foundedness of R. This frame condition is essentially second-order: no first-order formula can express it. Another well-known non-first-order definable modal axiom is the McKinsey Axiom $\Box\Diamond p \to \Diamond\Box p$. So we are confronted by an interesting 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 an obvious question: is there anything systematic about this? That is, are there any criteria for demarcating essentially first-order modal formulas from genuinely second-order ones?

5.3 First-order definable modal axioms

Before answering this question, let us say a little more about first-order definable modal axioms. For a start, these include many of the modal axioms one is likely to encounter in practice. As we've just seen, the characteristic axioms of the systems **T** and **K4** are first-order, and so are many of the axioms of newer systems arising in applications. For example, recall that in Section **??** we said that a binary modality $\Diamond(\varphi, \psi)$ might be viewed as describing a ternary composition relation for state transitions or symbol sequences. But if we are serious about this interpretation we should demand that composition be *associative,* that is, that the following first-order frame condition holds:

$$\forall xyzu((Rxyz \land Rzuv) \to \exists s(Rsyu \land Rxsv)).$$

And it turns out that there is a simple modal formula that corresponds to this:

$$\Diamond(\varphi, \Diamond(\psi, \theta)) \to \Diamond(\Diamond(\varphi, \psi), \theta).$$

Secondly, modal axioms φ corresponding to first-order frame conditions have the following pleasant property:

If a modal formula φ defines a first-order frame condition α , then the set of modal consequences of φ is recursively enumerable.

The reason is that a modal formula ψ is true on frames for φ iff its standard translation $ST_x(\psi)$ is true in all models of the first-order formula α. Thus first-order definable modal logics can draw on techniques from first-order theorem proving, and standard first-order model theory is available for their semantic analysis.

So let's turn to the systematicity issue. Upon closer inspection, first-order conditions often turn out to be computable from the shape of the given modal axiom — for example the quantifier shape of the first-order formula for transitivity is matched precisely by the sequence of modal boxes in the **K4**-axiom. A key result explaining these correspondences is the

Theorem 5.3 (Sahlqvist Correspondence Theorem) *There is an effective method for computing first-order equivalents for modal axioms $\varphi \to \psi$ 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.*

The proof proceeds by substituting first-order descriptions of 'minimal syntactic values' for proposition symbols that validate the antecedent of the modal axiom. For example, the **K4** axiom $\Box p \to \Box\Box p$ has a first-order standard translation of the form

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

A minimal valuation for p making the antecedent true is $Pu := Rxu$. Substituting this description for the unary predicate P and dropping the then tautologically true antecedent, we are left with a consequent formula

$$\forall y(Rxy \to \forall z(Ryz \to Pz)),$$

involving only the relation R, which is precisely first-order transitivity. By similar considerations about minimal values (combined with some pulling out of diamonds to become universal prefix quantifiers) the algorithm will show, for example, that the ubiquitous modal **.2** axiom corresponds to the well known relational property of confluence:

$$\mathfrak{F}, s \models \Diamond\Box p \to \Box\Diamond p \text{ iff } \mathfrak{F}, s \models \forall xy(Rxy \to \forall z(R \to \exists u(Ryu \land Rzu))).$$

The Sahlqvist Theorem and its proof method are very powerful: it applies to multi-modal languages with arbitrary arity modalities. Nevertheless there are also first-order definable modal axioms that do not fall under the method. The **K4.1** axiom

$$(\Box p \to \Box\Box p) \land (\Box\Diamond p \to \Diamond\Box p)$$

is a conjunction of the **K4** axiom with the McKinsey axiom. It defines the frames with a transitive relation where every point has a successor. But this first-order equivalence cannot be computed using the substitution method (van Benthem 1985).

Are there other general things we can say about the modal formulas that give rise to first-order definable axioms? Here is a semantic characterization (again from van Benthem 1985) that uses some elementary model theory: *A modal formula defines a first-order frame property iff it is preserved under taking ultrapowers of frames*. Still, this is a abstract feature, and it is not easy to use it to recognise whether a given modal formula is first-order over frames. In fact, the problem of determining whether a modal formula expresses a first-order condition on frames turns out to be undecidable (Chagrova 1985).

Many formulas that violate the syntactic constraints demanded by the Sahlqvist Theorem turn out to be non-first-order; a notable example is the McKinsey axiom. Results like this are proved by showing that the modal axiom in question lacks some typical first-order property, such as preservation under ultraproducts, or Löwenheim-Skolem properties. Most of these results have been studied in depth only for the basic modal language, which has served as sort of mathematical laboratory for the model theory of modal languages and logics. But there are some extensions to richer formalisms.

5.4 Correspondence in richer languages: fixed-point extensions

The substitution algorithm runs into difficulties with more complex antecedents. Consider Löb's axiom $\Box(\Box p \to p) \to p$. The point of computing minimal antecedent values in Sahlqvist axioms $\varphi \to \psi$ is this. Firstly, a Sahlqvist antecedent φ is true under any value for its proposition symbols iff it is true under their *minimal* values. Secondly, the latter minimal predicates are first-order definable. Now, the Löb antecedent does not support the second part of this analysis. But this does not mean that all that can be said here is that the Löb's axiom is second-order definable — for there is indeed a smallest semantic value for the predicate P which will make the Löb 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 this antecedent has a special syntactic form. Call a first-order formula $\varphi(P)$ *intersective* if it is one of the forms:

(i) $\forall x(\varphi(P, Q, x) \to Px)$, with P occurring only positively in $\varphi(P, Q, x)$.

(ii) $\psi(P, Q)$,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 *minP.*$\varphi(P)$, the *minimal* satisfying predicate. 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(FO)*, first-order logic with *monotonic fixed-points*. *LFP(FO)* 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.

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 we obtain an equivalent in *LFP(FO)*. As an illustration, 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 Axiom can be computed to be in *LFP(FO)*. Of course, in this particular case this is already known independently, for this is just the property of well-foundedness. But the method given here works more generally. For example, consider the less well known modal axiom of *cyclic return*:

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

Again this fails the Sahlqvist criterion. But again, the antecedent is intersective, 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.

This is the beginning of a further layering of modal axioms with respect to semantic complexity. For there are also modal formulas with frame equivalents not even in *LFP(FO)*. 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. Indeed (van Benthem 2003) also has a purely modal example — but the most obvious candidate for non-*LFP(FO)*-ness is the McKinsey Axiom, whose antecedent is typically non-intersective.

5.5 *Modally definable frame classes*

For a given modal axiom one can ask what kind of frame property it defines. But, conversely, one can also ask whether given frame properties are definable by means of modal formulas. Of the several basic results of this sort, we mention the following result due to Goldblatt and Thomason 1974. It characterises the modally definable first-order frame properties in terms of their special semantic behaviour.

Theorem 5.4 (Goldblatt-Thomason Theorem) *A first-order frame property is modally definable iff it is preserved under taking generated subframes, p-morphic frame images, disjoint unions, and inverse ultrafilter extensions.*

Again, this is an abstract characterization, obtained using a modal version of the Stone Representation Theorem plus the Birkhoff Theorem from Universal Algebra. It is not known which syntactic first-order forms of the frame condition $\alpha(R)$ are necessary and sufficient to guarantee this behaviour.

5.6 *First-order logic as modal logic*

We said at the beginning that a broad landscape of possible modal logics (note the plural) is a typical feature distinguishing modal logic from standard first-order logic. But then we suggested that there may be just one true modal logic, the minimal logic K, whereas the others are more properly viewed as special theories with additional axioms. After all, we would call the theory of linear order the 'predicate logic of linearity'. Now let's turn the tables. We can also take a modal look at first-order logic itself, and then find that it becomes such a specialised theory itself! To see this, recall the truth condition for the existential quantifier:

$$M, s \models \exists\varphi \text{ iff there exists a } d \text{ in } D^M \text{ such that } M, s[x := d] \models \varphi$$

This has the familiar pattern for evaluating an existential modality $\langle x \rangle$

$$M, s \models \langle x \rangle\varphi \text{ iff there exists a } t \text{ such that } R^x st \text{ and } M, t \models \varphi$$

Thus a first-order becomes a modal universe of 'states', the usual variable assignments, which are related by accessibility relations for individual variables:

$$R^x st \text{ iff } s(x) = t(y) \text{ for all variables } y \text{ distinct form } X$$

Now the usual validities of first-order logic can be deconstructed into several layers. First, there is a decidable core consisting of the minimal modal logic, which contains such ubiquitous laws as monotonicity of first-order quantifiers:

$$\forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$$

This level makes no presuppositions whatsoever concerning the form of first-order models, which could have any kind of 'states' and 'variable shift relations' R^x. At the next level, we find laws recording universal effects of having states be variable assignments, with the special shift relation 'agreeing up to the value for x'. For example

$$\forall x \to \forall x\forall x\varphi$$

expresses the transitivity of R^x, and indeed, all the laws of **S5** hold here. Models of this sort need not contain all variable assignments, and such gaps can make some variables dependent on others in the way they can change their object values. The resulting logic, without blanket assumptions of variable independence is still decidable. Finally, most specifically, some first-order laws express *existence* properties requiring that there be enough states to create some pattern. Here is another well-known principle for reasoning with quantifiers. It resembles the earlier modal **.2** axiom.

$$\exists x\forall y\varphi \to \forall y\exists x\varphi$$

which expresses confluence: whenever $R^x st$ and $R^y su$ then there also exists a state v such that $R^y tv$ and $uR^x v$.

Thus modal analysis reveals unexpected fine-structure inside the apparently monolithic class of 'standard validities' of first-order logic: they can be valid for different geometrical reasons. Summing up, we get a highly unorthodox view. the modal core of standard logic is decidable — and the usual undecidability of first-order logic just means piling up special existential model conditions to make state sets behave so much like full function spaces over the model domain that their logic encodes enough mathematics to become undecidable.

6 Richer languages

The purpose of this section is to discuss a typical, but not yet widely appreciated, aspect of contemporary modal logic: flexible language (re-)design. As we have seen, the basic modal language has a number of attractive properties, and as the bisimulation invariant fragment of the first-order correspondence language it is a special tool when it comes to talking about graphs. Nonetheless, many of its design parameters were fixed by historical accident. Perhaps judicious experimentation could lead to improvements, or at least to interesting variants? Modal logicians have been carrying our such experiments for years, and in this section we survey some of their work.

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 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 using the following satisfaction clause:

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

Now, such n-place modalities are undeniably useful for certain purposes, but developing their theory (standard translation, bisimulation, and so on) is essentially a matter of sprinkling our earlier work with additional indices. These operators don't give rise to richer languages in any logically interesting sense.

As we shall see, the richer languages explored in this section offer more. Moreover, their richness arises from different sources. 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 even enrich by abandoning modal syntax and using first-order syntax instead!

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.

6.1 The universal modality

Let's start by feeding 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 task, 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 TBox and ABox, acknowledges the importance of this problem (the information in TBoxes has to be globally satisfied, while the information in ABoxes only has to be locally satisfied). But this architectural distinction is not reflected in the object language, 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:

$$\mathfrak{M}, w \models E\varphi \text{ iff there is a } u \in W \text{ such that } \mathfrak{M}, u \models \varphi$$

$$\mathfrak{M}, w \models A\varphi \text{ iff for all } u \in W \text{ we have } \mathfrak{M}, u \models \varphi.$$

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 metatheoretic 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 [7] 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 6.1 *The satisfiability problem for the basic modal language enriched with the universal modality is EXPTIME-complete.*

Proof. See Hemaspaandra [24], or her earlier PhD thesis Spaan [44]. $\dashv$

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:

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

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 $\forall x \exists y\, Ryx$ 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. For the following result is the Goldblatt-Thomason Theorem for the basic modal language, with closure under disjoint unions and generated subframes stripped away.

Theorem 6.2 *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 [21]. $\dashv$

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, $\Box(\Box p \to p) \to p)$ 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:

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

(This example is from Goranko and Passy [21], 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 [42] 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 setting and, like the universal modality, has a smooth metatheory; see de Rijke [11] 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 *identical* states. 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 Arther Prior [39,40] in the 1960s: sort the propositional symbols, and use *formulas as terms*. Let's do this right away. Take a language of basic modal logic (with propositional symbols p, q, r, and so on) and add a second sort of propositional symbol. The new symbols are called *nominals*, and are typically written i, j, k, and l. Both types of propositional 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:

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

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:

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

This is identical to the preceding formula, except that we have replaced the propositional 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:

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

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

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

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 nodalities, for all instances of $@_i\varphi \leftrightarrow \neg @_i \neg\varphi$ are valid. So we are free to regard satisfaction operators are 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 it:

$$\begin{array}{rl}
\forall x \neg Rxx & i \to \neg\Diamond i \\
\forall x \forall y(Rxy \to \neg Ryx) & i \to \neg\Diamond\Diamond i \\
\forall x \forall y(Rxy \wedge Ryx \to x = y) & i \to \Box(\Diamond i \to i) \\
\forall x \forall y(Rxy \vee x = y \vee Ryx) & @_j\Diamond i \vee @_j i \vee @_i \Diamond j.
\end{array}$$

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:

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

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 3.1 to make it suitable for the basic hybrid language and for the constant-enriched first-order correspondence language:

Definition 6.3 (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 points (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 Z.*

Lemma 6.4 (Bisimulation-with-names Invariance Lemma) *If E is a bisimulation-with-names between $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$, and wEw', then w and w' satisfy the same basic hybrid formulas.*

Proof. An easy extension of the inductive proof of Lemma 3.5. There are only two new cases to check. $\dashv$

And now for the key result:

Theorem 6.5 (Hybrid Characterisation Theorem) *The following are equivalent for all first-order formulas $\varphi(x)$ in at most one free variable x:*

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

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

Proof. That clause *(ii)* implies *(i)* is a more or less immediate consequence of Lemma 6.4. The hard direction is showing that clause *(i)* implies *(ii)*. The original proof can be found in Areces, Blackburn and Marx [3]. $\dashv$

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 hybridization 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 6.6 *The satisfiability problem for the basic hybrid language over arbitrary models is PSPACE-complete.*

Proof. See Areces, Blackburn and Marx [3]. $\dashv$

For a detailed overview of hybrid logic, see Chapter **??** 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 [30], 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 [19] observed that Until offers precisely what is required to state *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 is a key component of 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 from a computer science perspective, see Clarke, Grumberg and Peled [9].

Now, we briefly met the Until operator in Section 4 when we discussed model checking. There we placed a restriction on the relations that could interpret it (we insisted on working with the transitive closure of an

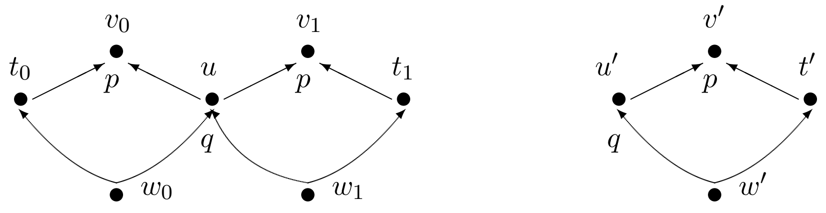

Fig. 15. Until not definable in basic modal logic

irreflexive relation). Here we drop this requirement and define Until and Since in their most general form:

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

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 15; 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. This result can be strengthened. Even if we restrict ourselves to linear models, the basic modal language can't define Until (see Proposition 7.10 in [7] for a proof that it can't even do so on the real numbers).

So adding S and U to the basic modal language yields new expressivity — but how much? We shall state 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.

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:

$$\begin{aligned} ST_x(U(\varphi, \psi)) &= \exists z\, (Rxz \land ST_z(\varphi) \land \forall y\, (Rxy \land Ryz \to ST_y(\psi))) \\ ST_x(S(\varphi, \psi)) &= \exists z\, (Rzx \land ST_z(\varphi) \land \forall y\, (Rzy \land Ryx \to ST_y(\psi))). \end{aligned}$$

(Incidentally, observe that we need three variables to specify this translation. Thus the translation doesn't give us an easy decidability result for the enriched language, though in fact its satisfiability problem is decidable over arbitrary models.)

So what does Kamp's theorem say? First some preliminary definitions. Let K be a class of models. We say that a modal language is *expressively complete over* K, 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 K). 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 order are the real numbers $(\mathbb{R}, <)$ and the natural numbers $(\mathbb{N}, <)$ under their usual orderings. And now we have:

Theorem 6.7 (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 [30]. Elegant modern proofs (and proofs of related results) can be found in Gabbay, Hodkinson and Reynolds [18]. $\dashv$

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, in 1997, Kurtonina and de Rijke [33] gave a definition which enabled a Characterisation Theorem 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 uses of 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 attempt 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 [41]).

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 most readings 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 $\mathfrak{M} = \langle W, C, V\rangle$. Here W is a set of 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 "world u has more in common with world w than world v does". It is standard to demand that C satisfies $\forall uvz(C_w uv \wedge C_w vz \to C_w uz)$, 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" [10]. It fits in well with our intuitions (at least on finite models):

$$\mathfrak{M}, w \models \varphi > \psi \text{ iff all } \textit{minimal } \varphi\text{-worlds in the } w\text{-centred ordering } C_w uv \text{ are } \psi \text{ worlds.}$$

Note that φ-minimal worlds around w are the only ones we consider. That is, this satisfaction definition is not given purely in terms of simple frame conditions (such as the "inspect the R-successor states" familiar from the basic language) it also takes into account which formulas are true and where. 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 left-monotonicity.

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 thing readable, we shall write use $\varphi(v)$ as shorthand for $\mathfrak{M}, v \models \varphi$, and similarly for ψ):

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

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 φ. 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:

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

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 [8] and Veltman [49], 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 6.8 *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 (that is, no worse than propositional logic) for formulas with bounded nesting of conditionals. If uniformity is assumed (that is, if we assume that all worlds agree on what worlds are possible) the complexity rises to EXPTIME-complete, even for formulas with bounded nesting. If absoluteness is assumed (that is, if we assume that all worlds agree on all conditional statements) the decision problem is NP-complete for formulas with arbitrary nesting.*

Proof. See Friedman and Halpern [17]. $\dashv$

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 $>$ is 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's [47] survey of correspondence theory examines conditional axioms corresponding to the triangle inequalities of 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 [20]). 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"; the approach was first introduced and explored in Ryan and Schobbens [43].

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 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:

$$\mathrm{ST}_x(\Diamond\varphi) = \exists y(Rxy \wedge \mathrm{ST}_y(\varphi))$$
$$\mathrm{ST}_x(\Box\varphi) = \forall y(Rxy \to \mathrm{ST}_y(\varphi)).$$

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 [2]. Guarded formulas φ are built up as follows:

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

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 guard, is a predicate symbol too. The key point to observe is that the free variables of φ are also free in the guard. The set of all guarded first-order formulas is called the guarded fragment.

Theorem 6.9 *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 [22] 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 [1]. $\dashv$

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 [6]) and the packed fragment (see Marx [35]) have been proposed which do precisely this. Let's take a quick look at the packed fragment.

The packed fragment allows us to use allow *composite guards* γ instead of merely atomic guards G: guards are now conjunctions of the following kinds of formulas:

$$x_i = x_j \text{ or } R(x_{i_1},\ldots,x_{i_n}) \text{ or } \exists x_{j_1}\ldots\exists x_{j_m}\,R(x_{i_1},\cdots,x_{i_n}) \text{ or } \forall x_{j_1}\ldots\forall x_{j_m}\,R(x_{i_1},\cdots,x_{i_n}).$$

The crucial point, however, is to state some restriction on the way we quantify variables to ensure that decidability is retained. In the packed fragment we do this as follows. We say that a guard φ is a *packed guard* if for every pair of distinct free variable x_i and x_j it contains, there is a conjunct in φ in which x_i and x_j both occur free. Then packed formulas are built up as follows:

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

where γ is a packed guard, φ is a packed formula, and (as with the guarded fragment) all variables free in φ are free in γ. The set of all packed first-order formulas is called the packed fragment.

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:

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

The packed fragment is also computationally well-behaved:

Theorem 6.10 *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 [22]. The original proof of the finite model property for the packed fragment (and the loosely guarded fragment) can be found in Hodkinson [25]; a more elegant proof can be found in Hodkinson and Otto [26]. $\dashv$

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 'existential' ones (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 sublogic. 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 [2]. But subsequent work has shown that the picture is somewhat mixed. There is indeed a natural notion of 'guarded bisimulation' (again see [2]) which characterises the guarded fragment as fragment of first-order logic. Moreover, Beth definability holds (see Hoogland, Marx and Otto [28]). 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 [27]).

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 $\exists\forall$ 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 [12]) show that first-order logic is badly behaved when interpreted over such structures. Nowadays it is routine to extend first-order languages with second-order constructs (such as the ability to take transitive closure or form fix-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 [37], and the language now called PDL was first investigated by Fisher and Ladner [15,16]. 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 execution of π from the current state leads to a state with the information φ". Crucially, the inductive structure of the programs is made explicit in PDL's syntax. Complex programs are built out of basic programs using four program constructors, and the diamonds reflect this. 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 formulas, $\langle \pi_1 \cup \pi_2 \rangle$, $\langle \pi_1 \,;\, \pi_2 \rangle$, $\langle \pi_1^* \rangle$ and $\langle \varphi ? \rangle$ are diamonds too. Note the unusual syntax of the test constructor diamond: it makes a modality out of a formula (incidentally, 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. 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 is simply

$$R^{\varphi ?} \;=\; \{(x, y) \mid x = y \text{ and } y \models \varphi\}.$$

This makes sense in any model; no additional frame conditions need to be imposed here. But the three remaining constructors certainly *do* demand additional frame structure. Here's what is required:

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

These restriction 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 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.

What can we say with PDL? At the level of models we can express some familiar programming constructs:

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

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 where $R^{\pi_1 \cup \pi_2} = R^{\pi_1} \cup R^{\pi_2}$, and the formula

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

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 we 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

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

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

Theorem 6.11 *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 [15,16]. The existence of an EXPTIME algorithm for PDL satisfiability was proved in Pratt [38]. ⊣

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 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 an 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 [31].

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 propositional variable (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 propositional 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 the satisfaction definition for the basic modal language:

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

Furthermore, for any propositional 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 propositional variable 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 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 propositional variables. That is, we shall write expressions like $\mu p.\varphi$, in which all free occurrence of the propositional variable p in φ are bound by the 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 propositional variable when all free occurrences of the variable occur positively. With this restriction in mind we define:

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

That is, the set assigned to $\mu p.\varphi$ is the least fixed-point (as specified by the Knaster-Tarski Theorem) 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

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

(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:

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

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:

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

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. We do so as follows:

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

In fact the modal μ-calculus, is strictly more expressive than PDL. The simplest example of 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 is can be expressed in 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 modal μ-calculus. For remarks and references on this topic, see Chapter **??** 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 6.12 *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 Kozen and Parikh [32]. The finite model property was first established in Street and Emerson [45]. The complexity result is from Emerson and Jutla [13]. $\dashv$

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

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

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

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. Thus the modal μ-calculus can be viewed as a fragment of monadic second-order logic. Which fragment? This one:

Theorem 6.13 *The modal μ-calculus is the bisimulation invariant fragment of monadic second-order logic.*

Proof. See Janin and Walukiewicz [29]. $\dashv$

For more on the modal μ-calculus, see Chapter **??** 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 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 [50] and Prior [39,40], and the idea was central to the work of the 'Sofia School' (see, for example, Passy and Tinchev [36] 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). Now, there have been various attempts to find general patterns explaining which parts of first-order logic are involved in 'modal' languages. Gabbay 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 uniform model checking complexity is in PTIME (see Vardi [**?**]) as opposed to PSPACE for the full first-order language. On the other hand, 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 Vardi [48] "Why is modal logic so robustly decidable?"), which emphasises the semantic adequacy of tree-like models obtainable via bisimulation unravelling of arbitrary graph models. This type of explanation transcends first-order logic, but it does not provide concrete syntactic explanation. For the latter, the current best explanation is that provided by the guarded fragment and it relatives (which are, arguably, the strongest known modal languages).

As we saw, such fragments locate the essence of modal logic in the *restriction* on the quantification performed by the modalities. One attractive property of this analysis it its logical resilience: it turns out that it extends beyond the setting of first-order enrichments to second-order enrichment (something that was not forseen when the guarded fragment was first isolated). A striking example is the result in Grädel and Walukiewicz [23] that the extension of the guarded fragment with the fixed-point operators μ and ν remains decidable. By way of contrast, validity for the full first-order logic extended with these operators in 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 logic: often modal fragments can bear the weight of additional higher-order constructions (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 [**?**] and Baltag [**?**])for investigating non-well founded set theory 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, and moreover a modal invariance 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 [5]); do such results hold for 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 [14] for some early results and ten Cate [46] for a more recent discussion. The equation "modality = guarding" should be simultaneously be 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. One such approach is to apply ideas form abstract model theory (see Barwise and Feferman [4]). This was first done in de Rijke [42], who proved a modal analog of Lindström's [34] celebrated characterisation of first-order logic, in which bisimulation replaced Lindström's use of isomorphism. More recently, ten Cate [46] has use the approach to search for general results on which classes of modal logics can have such properties as interpretation. Perhaps such investigations do not tell us what the space of modal logics actually is, but they give us a clearer idea of what is out there.

7 New descriptive challenges

Traditional motivations for and applications of modal logic came from philosophy, and the study of such topics as modality, knowledge, conditionals, and obligations. Some strands also concerned mathematics: witness modal logics of time, space, or provability. Gradually, in the 1960s and 1970s further influences arose that made the areas much 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 powers of agents in social choice theory. Many of these applications fit squarely into the perspective represented in this chapter. But some also raise technical issues of their own. For example, some logics make use of *weaker* language than the basic modal language, not extensions. And when (for example) the booleans are gone, modal logic becomes quite a different game.

7.1 An example where it all comes together: games

An interesting example which shows the issues in modern applications are games. Games are a natural continuation of the process view of model semantics, bringing it in line with the realities of modern computing, as they describe *interactive processes* between different agents. But they also bring in other strands from our presentation, in particular the epistemic stance, as they crucially involve the beliefs preferences and intentions of the players. Games have occurred so far int our presentation as tools for bringing out the interactive essence of key logical tasks, such as model checking or model comparison. Such games are called *logic games*, and the lead to interesting connections with game theory. For example, most logic games are two-player zero-sum games of finite depth, and Zermelo's theorem form 1913 then tells us that any such game is 'determined': one of the two players must have a winning strategy. For example, either Duplicator or Spoiler wins any given finite length model comparison game; there is not further option. But modal logic also applied to arbitrary games, for whatever purpose, providing so-called *game logics* that describe both external reasoning about and internal reasoning by players of a game. Chapter **??** of this handbook is devoted to this interface, but for our purposes here it may be useful to point out how much modal logic occurs with the single setting of a game. We restrict ourselves to so-called ‘extensive games’ — even though modal style analysis is also quite feasible for so-called ‘strategic games’ where one only records players entire strategies and the outcomes of playing them.

Modal logics of moves
For a start, an extensive game involves a tree of possible histories generated by moves available to players at their turns, while ending in final states where some pay-off may occur. As they stand, such structures are models for a basic modal language with several modalities. For example, assuming the Player 1 moves first, and then Player 2 the formula $[\textit{move-1}]\langle\textit{move-2}\rangle p$ says that Player 2 has guaranteed response to whatever Player 1 does which guarantees that p will occur. In game-theoretic terms, Player 2 has a *strategy* ensuring that p. With logier games, chains of modalities of the form $\Box\Diamond\Box\Diamond\Box\Diamond\cdots$ express the existence of various strategies.

Dynamic logic of strategies
To discuss these strategies more explicitly, note that a strategy for a player is a map to available moves — or in a more general setting, a relation constraining possible moves at relevant turns, An obvious formalism for defining such relations is PDL, and indeed the program constructors encountered there make immediate sense for games as well: strategies are iterative constructs out of conditional instructions of the form “if she plays this, then I play that”.

Fixed-point definitions of strategic equilibria
Nevertheless, general notions of game solution may go beyond PDL, and Zermelo’s theorem is an example. Defining the two mutually exclusive predicate “Player 1 has a winning strategy]] and “Player 2” has a winning strategy as properties of nodes in a game tree requires inductive definition in the modal μ-calculus, And the same fixed-point language is needed for more complex notions of game solution that involve numerical utilities, such as the Nash equilibrium.

Strategies once more, threats, and conditonal logic
A strategy is a rule telling us what what to do under all circumstances, even those that do not actually occur. My response is prescribed for every move you could play, even though a single run of the game only allows one move by you. And even more mysteriously, my strategy will lead to just a part of the game tree, while forgoing others; but even so, the strategy also prescribes moves at my turns in those ‘inaccessible parts’! This is relevant to threats, where you might reason about what would happen were I to act differently from my benevolent current intention, This counterfactual character of strategies (emphasized in Stalnaker (199X) [**?**]) naturally brings in statements of the form “if a were to happen, then I would play b”. Hence, a really full-fledged modal logic of strategies also involves conditional logic.

Knowledge and rationality
But possible moves and strategies are not the whole story. Even games of perfect information, where players know all relevant moves, and can observe every relevant fact that happens, involve players’ reasoning about what others will do, and their uncertainty about future moves, We cannot predict how the game will unfold, unless we make assumptions about the capacities and rational intentions of other players. This brings in issues of knowledge and belief, which have infiltrated game theory since the 1970s when Aumann independently rediscovered Hintikka’s epistemic logic. As an example of such reasoning, even standard game-theoretic solution algorithms such as ‘backward Induction’ or ‘Iterated Removal of Strictly Dominated Strategies’ only

make sense by making various epistemic assumptions that can be brought to light in suitable modal languages (Aumann, de Bruinn 2004), van Benthem 2002).

Preference logics
Most game logics so far have emphasized available moves and strategies, and the 'control' players have over outcomes, or just courses, of the game. But the heart of any rationality analysis is what players will do given the utilities they attach to outcomes. Alternatively, we can assume that players have preference relations allowing them to compare various outcomes. Making the latter structure explicit again requires modal languages, this time with modalities accessing binary or ternary preference relations (see Harrenstein van der Hoek and Wooldrige 200x, van Benthem, van Otterloo and Roy 2005 for fully explicit characterisations of game-theoretic equilibria in such formalisms.

Imperfect informaition
So far, uncertainty played a role in what players know or believe about the future course of a game which is fully transparent by itself. But many games involve 'imperfect information'. For example, in a card game we typically do not see each player's hand, and it is this limited observational capacity plus our powers of reasoning and anticipation that drive the whole process. Games of imperfect information again involve epistemic logic, this time also in the form of uncertainties about player's exact position in the game tree. Thus by combining knowledge and action modalities we can express typical game-ttheoretic quandaries, such as

$$K(\langle a\rangle \vee \langle b\rangle p) \wedge \neg K\langle a\rangle p \wedge \neg K\langle b\rangle p.$$

I know that I have a move (a or b) allowing me to achieve p, but I do not know which one.

Dynaics: update and revision
Finally, with all this descriptive apparatus in place to describe the static structure of a game at successive stages of its development, there is still the issue of explicit dynamics as the game moves forward. For example, in a card game, we *update* our information in a systematic manner as cards fall on the table. Saying just how this happens requires dynamic-epistemic logics of information update. Moreover, as a game proceeds, surprising things may happen that lead us to question our earlier assumptions about other players — instead of a hard-nosed egotist we may observe a gentle altruist at work. If that happens we need to revise current beliefs about the development of the game, and belief revision comes in — which fits in with the earlier-mentioned conditional logic of strategies.

This the concrete phenomenon of games that we are all familiar with involves just about every strand of modal logic that we have seen — and these strands need to be put together in delicate ways.

References

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

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

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

[4] Barwise, J. and S. Feferman, "Model-theoretic logics," Springer, 1985.

[5] Barwise, J. and J. van Benthem, *Interpolation, preservation, and pebble games*, jsl **29** (1999), pp. 881–903.

[6] Benthem, J. v., *Dynamic bits and pieces*, Technical Report LP-97-01, Institute for Logic, Language and Computation, University of Amsterdam (1997).

[7] Blackburn, P., M. de Rijke and Y. Venema, “Modal Logic,” Cambridge University Press, 2001.

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

[9] Clarke, E., O. Grumberg and D. Peled, “Model Checking,” MIT Press, 1999.

[10] Counterfactuals, “D. Lewis,” Blackwell, 1973.

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

[12] Ebbinghaus, H.-D. and J. Flum, “Finite Model Theory,” Perspectives in Mathematical Logic, Springer, 1995.

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

[14] Fine, K., *Propositional quantifiers in modal logic*, Theoria **36** (1970), pp. 331–346.

[15] Fischer, M. and R. Ladner, *Propositional modal logic of programs*, in: *9th ACM Sympos. Theory of Comput.*, 1977, pp. 286–294.

[16] Fischer, M. and R. Ladner, *Propositional dynamic logic of regular programs*, Journal of Computer and System Sciences **18** (1979), pp. 194–211.

[17] Friedman, N. and J. Halpern, *On the complexity of conditional logic*, in: *Proceedings of the 4th International Conference on Principles of Knowledge Representation (KR ’94)*, 1994, pp. 202–213.

[18] Gabbay, D., I. Hodkinson and M. Reynolds, “Temporal Logic: Mathematical Foundations and Computational Aspects,” Oxford University Press, 1994.

[19] Gabbay, D., A. Pnueli, S. Shelah and J. Stavi, *On the temporal analysis of fairness*, in: *Proc. 7th ACM Symposium on Principles of Programming Languages*, 1980, pp. 163–173.

[20] Gärdenfors, P. and H. Rott, *Belief revision*, in: D. Gabbay and F. Guenthner, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, 4, Oxford University Press, 1995 pp. 325–408.

[21] Goranko, V. and S. Passy, *Using the universal modality: Gains and questions*, Journal of Logic and Computation **2** (1992), pp. 5–30.

[22] Grädel, E., *On the restraining power of guards*, Journal of Symbolic Logic **64** (1999), pp. 1719–1742.

[23] Grädel, E. and I. Walukiewicz, *Guarded Fixed Point Logic*, in: *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS ‘99, Trento*, 1999, pp. 45–54.

[24] Hemaspaandra, E., *The price of universality*, Notre Dame Journal of Formal Logic **37** (1996), pp. 174–203.

[25] Hodkinson, I., *Loosely guarded fragment of first-order logic has the finite model property*, Studia Logica **70** (2002), pp. 205–240.

[26] Hodkinson, I. and M. Otto, *Finite conformal hypergraph covers and Gaifman cliques in finite structures*, Bull. Symbolic Logic **9** (2003), pp. 387–405.

[27] Hoogland, E. and M. Marx, *Interpolation and definability in guarded fragments*, Studia Logica **70** (2002), pp. 373–409.

[28] Hoogland, E., 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*, LNAI **1705** (1999), pp. 273–285.

[29] Janin, D. and I. Walukiewicz, *On the expressive completeness of the propositonal mu-calculus with resepct to second order logic*, in: *Proceedings CONCUR ’96*, Lecture Notes in Computer Science **1119** (1996), pp. 263–277.

[30] Kamp, H., “Tense Logic and the Theory of Linear Order,” Ph.D. thesis, University of California, Los Angeles (1968).

[31] Kozen, D., *Results on the propostional mu-calculus*, Theoretical Computer Science **27** (1983), pp. 333–354.

[32] Kozen, D. and R. Parikh, *A decision procedure for the propositional mu-calculus*, in: *Proceedings of the 2nd Workshop on Logic of Programs*, LNCS **164** (1983), pp. 313–325.

[33] Kurtonina, N. and M. de Rijke, *Bisimulations for temporal logic*, Journal of Logic, Language and Information **6** (1997), pp. 403–425.

[34] Lindström, P., *On extensions of elementary logic*, Theoria **35** (1969), pp. 1–11.

[35] Marx, M., *Tolerance logic*, Journal of Logic, Language and Information **6** (2001), pp. 353–373.

[36] Passy, S. and T. Tinchev, *An essay in combinatory dynamic logic*, Information and Computation **93** (1991), pp. 263–332.

[37] Pratt, V., *Semantical considerations on Floyd-Hoare logic*, in: *Proc. 17th IEEE Symposium on Computer Science*, 1976, pp. 109–121.

[38] Pratt, V., *Models of program logics*, in: *Proc. 20th IEEE Symp. Foundations of Computer Science*, 1979, pp. 115–222.

[39] Prior, A., "Past, Present and Future," Clarendon Press, Oxford, 1967.

[40] Prior, A., "Papers on Time and Tense," Oxford University Press, 2003, New edition, edited by Hasle, Øhrstrom, Braüner, and Copeland.

[41] Restall, G., "An Introduction to Substructural Logics," Routledge, 2000.

[42] Rijke, M., "Extending Modal Logic," Ph.D. thesis, ILLC, University of Amsterdam (1993).

[43] Ryan, M. and P. Schobbens, *Counterfactuals and updates as inverse modalities*, Journal of Logic, Language and Information **6** (1997), pp. 123–146.

[44] Spaan, E., "Complexity of Modal Logics," Ph.D. thesis, ILLC, University of Amsterdam (1993).

[45] Street, R. and E. Emerson, *An automata theoretic decision procedure for the propostional mu-calculus*, Information and Computation **81** (1989), pp. 249–2644.

[46] ten Cate, B., "Model theory for extended modal languages," Ph.D. thesis, Institute for Logic, Language and Computation, University of Amsterdam (2004).

[47] van Benthem, J., *Correspondence theory*, in: D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, 3, Kluwer Academic Publishers, 2001 pp. 325–408.

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

[49] Veltman, F., "Logics for Conditionals," Ph.D. thesis, University of Amsterdam (1985).

[50] Wright, G. v., "An Essay in Modal Logic," North-Holland Publishing Company, 1951.