1

Basic Concepts

Languages of propositional modal logic are propositional languages to which sentential operators (usually called *modalities* or *modal operators*) have been added. In spite of their syntactic simplicity, such languages turn out to be useful tools for describing and reasoning about *relational structures*. A relational structure is a non-empty set on which a number of relations have been defined; they are widespread in mathematics, computer science, artificial intelligence and linguistics, and are also used to interpret first-order languages.

Now, when working with relational structures we are often interested in structures possessing certain properties. Perhaps a certain transitive binary relation is particularly important. Or perhaps we are interested in applications where 'dead ends,' 'loops,' and 'forkings' are crucial, or where each relation is a partial function. Wherever our interests lie, modal languages can be useful, for modal operators are essentially a simple way of accessing the information contained in relational structures. As we will see, the *local* and *internal* access method that modalities offer is strong enough to describe, constrain, and reason about many interesting and important aspects of relational structures.

Much of this book is essentially an exploration and elaboration of these remarks. The present chapter introduces the concepts and terminology we will need, and the concluding section places them in historical context.

Chapter guide

Section 1.1: Relational Structures. Relational structures are defined, and a number of examples are given.

Section 1.2: Modal Languages. We are going to talk about relational structures using a number of different modal languages. This section defines the basic modal language and some of its extensions.

Section 1.3: Models and Frames. Here we link modal languages and relational structures. In fact, we introduce *two* levels at which modal languages can be used to talk about structures: the level of *models* (which we explore in Chapter 2) and the level of *frames* (which is examined in Chapter 3). This section contains the fundamental *satisfaction definition*, and defines the key logical notion of *validity*.

Section 1.4: General Frames. In this section we link modal languages and relational structures in yet another way: via *general frames*. Roughly speaking, general frames provide a third level at which modal languages can be used to talk about relational structures, a level intermediate between those provided by models and frames. We will make heavy use of general frames in Chapter 5.

Section 1.5: Modal Consequence Relations. Which conclusions do we wish to draw from a given a set of modal premises? That is, which *consequence relations* are appropriate for modal languages? We opt for a *local* consequence relation, though we note that there is a *global* alternative.

Section 1.6: Normal Modal Logics. Both validity and local consequence are defined *semantically* (that is, in terms of relational structures). However, we want to be able to generate validities and draw conclusions *syntactically*. We take our first steps in modal proof theory and introduce Hilbert-style axiom systems for modal reasoning. This motivates a concept of central importance in Chapters 4 and 5: *normal modal logics*.

Section 1.7: Historical Overview. The ideas introduced in this chapter have a long and interesting history. Some knowledge of this will make it easier to understand developments in subsequent chapters, so we conclude with a historical overview that highlights a number of key themes.

1.1 Relational Structures

Definition 1.1 A *relational structure* is a tuple $\mathfrak{F}$ whose first component is a non-empty set W called the *universe* (or *domain*) of $\mathfrak{F}$, and whose remaining components are relations on W. We assume that every relational structure contains at least one relation. The elements of W have a variety of names in this book, including: *points*, *states*, *nodes*, *worlds*, *times*, *instants* and *situations*. $\dashv$

An attractive feature of relational structures is that we can often display them as simple pictures, as the following examples show.

Example 1.2 *Strict partial orders* (SPOs) are an important type of relational structure. A *strict partial order* is a pair (W, R) such that R is *irreflexive* ($\forall x\, \neg Rxx$) and *transitive* ($\forall xyz\, (Rxy \wedge Ryz \rightarrow Rxz)$). A strict partial order R is a *linear order* (or a *total order*) if it also satisfies the *trichotomy* condition: $\forall xy\, (Rxy \vee x = y \vee Ryx)$.

An example of an SPO is given in Figure 1.1, where $W = \{1, 2, 3, 4, 6, 8, 12, 24\}$

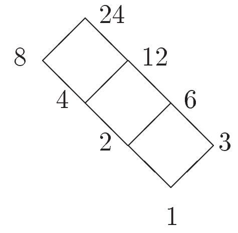


Fig. 1.1. A strict partial order.

and Rxy means 'x and y are different, and y can be divided by x.' Obviously this is *not* a linear order. On the other hand, if we define Rxy by 'x is numerically smaller than y,' we obtain a linear order over the same universe W. Important examples of linear orders are $(\mathbb{N}, <)$, $(\mathbb{Z}, <)$, $(\mathbb{Q}, <)$ and $(\mathbb{R}, <)$, the *natural numbers*, *integers*, *rationals* and *reals* in their usual order. We sometimes use the notation $(\omega, <)$ for $(\mathbb{N}, <)$.

In many applications we want to work not with *strict* partial orders, but with plain old *partial orders (*POs*)*. We can think of a partial order as the reflexive closure of a strict partial order; that is, if R is a strict partial order on W, then $R \cup \{(u,u) \mid u \in W\}$ is a partial order (for more on reflexive closures, see Exercise 1.1.3). Thus partial orders are transitive, *reflexive* ($\forall x\, Rxx$) and *antisymmetric* ($\forall xy\, (Rxy \wedge Ryx \to x = y)$). If a partial order is *connected* ($\forall xy\, (Rxy \vee Ryx)$) it is called a *reflexive linear order* (or a *reflexive total order*).

If we interpret the relation in Figure 1.1 reflexively (that is, if we take Rxy to mean 'x and y are equal, or y can be divided by x') we have a simple example of a partial order. Obviously, it is not a reflexive *linear* order. Important examples of reflexive linear orders include $(\mathbb{N}, \leq)$ (or $(\omega, \leq)$), $(\mathbb{Z}, \leq)$, $(\mathbb{Q}, \leq)$ and $(\mathbb{R}, \leq)$, the *natural numbers*, *integers*, *rationals* and *reals* under their respective 'less-than-or-equal-to' orderings. $\dashv$

Example 1.3 *Labeled Transition Systems* (LTSs), or more simply, *transition systems*, are a simple kind of relational structure widely used in computer science. An LTS is a pair $(W, \{R_a \mid a \in A\})$ where W is a non-empty set of states, A is a non-empty set (of *labels*), and for each $a \in A$, $R_a \subseteq W \times W$. Transition systems can be viewed as an abstract model of computation: the states are the possible states of a computer, the labels stand for programs, and $(u, v) \in R_a$ means that there is an execution of the program a that starts in state u and terminates in state v. It is natural to depict states as nodes and transitions R_a as directed arrows.

In Figure 1.2 a transition system with states w_1, w_2, w_3, w_4 and labels a, b, c is shown. Formally, $R_a = \{(w_1, w_2), (w_4, w_4)\}$, while $R_b = \{(w_2, w_3)\}$ and $R_c = \{(w_4, w_3)\}$. This transition system is actually rather special, for it is *deterministic*:

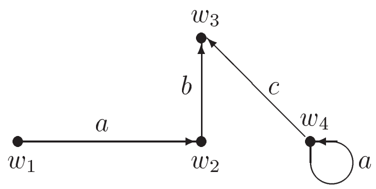


Fig. 1.2. A deterministic transition system.

if we are in a state where it is possible to make one of the three possible kinds of transition (for example, an a transition) then it is fixed which state that transition will take us to. In short, the relations R_a, R_b and R_c are all *partial functions*.

Deterministic transition systems are important, but in theoretical computer science it is more usual to take *non-deterministic* transition systems as the basic model of computation. A non-deterministic transition system is one in which the state we reach by making a particular kind of transition from a given state need not be fixed. That is, the transition relations do not have to be partial functions, but can be arbitrary relations.

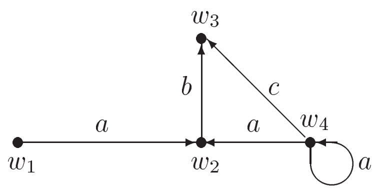


Fig. 1.3. A non-deterministic transition system.

In Figure 1.3 a non-deterministic transition system is shown: a is now a non-deterministic program, for if we execute it in state w_4 there are two possibilities: either we loop back into w_4, or we move to w_2.

Transition systems play an important role in this book. This is not so much because of their computational interpretation (though that is interesting) but because of their sheer ubiquity. Sets equipped with collections of binary relations are one of the simplest types of mathematical structures imaginable, and they crop up just about everywhere. $\dashv$

Example 1.4 For our next example we turn to the branch of artificial intelligence called knowledge representation. A central concern of knowledge representation is objects, their properties, their relations to other objects, and the conclusions one can draw about them. For example, Figure 1.4 represents some of the ways Mike relates to his surroundings.

One conclusion that can be drawn from this representation is that Sue has chil-

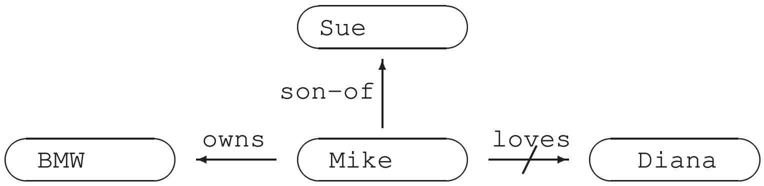


Fig. 1.4. Mike and others.

dren. Others are not so clear. For example, does Mike love Sue, and does he love his BMW? Assuming that absence of a `not_loves` arc (like that connecting the Mike and the Diana nodes) means that the loves relation holds, this is a safe conclusion to draw. There are often such 'gaps' between pictures and relational structures, and to fill them correctly (that is, to know which relational structure the picture corresponds to) we have to know which diagrammatic conventions are being assumed.

Let's take the picture at face value. It gives us a set $\{\texttt{BMW}, \texttt{Sue}, \texttt{Mike}, \texttt{Diana}\}$ together with binary relations `son-of`, `owns`, and `not_loves`. So we have here another labeled transition system. $\dashv$

Example 1.5 Finite trees are ubiquitous in linguistics. For example, the tree depicted in Figure 1.5 represents some simple facts about phrase-structure, namely that a sentence (S) can consist of a noun phrase (NP) and a verb phrase (VP); an NP can consist of a proper noun (PN); and VPs can consist of a transitive verb (TV) and an NP.

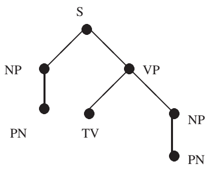


Fig. 1.5. A finite decorated tree.

Trees play an important role in this book, so we will take this opportunity to define them. We first introduce the following important concepts.

Definition 1.6 Let W be a non-empty set and R a binary relation on W. Then R^+, the *transitive closure* of R, is the smallest transitive relation on W that contains R. That is,

$$R^+ = \bigcap\{R' \mid R' \text{ is a transitive binary relation on } W \;\&\; R \subseteq R'\}.$$

Furthermore, R^*, the *reflexive transitive closure* of R, is the smallest reflexive and transitive relation on W containing R. That is,

$$R^* = \bigcap\{R' \mid R' \text{ is a reflexive transitive binary relation on } W \ \& \ R \subseteq R'\}. \quad \dashv$$

Note that R^+uv holds if and only if there is a sequence of elements $u = w_0$, w_1, $\ldots$, $w_n = v$ $(n > 0)$ from W such that for each $i < n$ we have Rw_iw_{i+1}. That is, R^+uv means that v is reachable from u in a finite number of R-steps. Thus transitive closure is a natural and useful notion; see Exercise 1.1.3.

With these concepts at our disposal, it is easy to say what a tree is.

Definition 1.7 A *tree* $\mathfrak{T}$ is a relational structure (T, S) where:

(i) T, the set of nodes, contains a unique $r \in T$ (called the *root*) such that $\forall t \in T\ S^*rt$.
(ii) Every element of T distinct from r has a unique S-predecessor; that is, for every $t \neq r$ there is a unique $t' \in T$ such that $St't$.
(iii) S is acyclic; that is, $\forall t \neg S^+tt$. (It follows that S is irreflexive.) $\dashv$

Clearly, Figure 1.5 contains enough information to give us a tree (T, S) in the sense just defined: the nodes in T are the displayed points, and the relation S is indicated by means of a straight line segment drawn from a node to a node immediately below (that is, S is the obvious *successor* or *daughter-of* relation). The root of the tree is the topmost node (the one labeled S).

But the diagram also illustrates something else: often we need to work with structures consisting of not only a tree (T, S), but a whole lot else besides. For example, linguists wouldn't be particularly interested in the bare tree (T, S) just defined, rather they'd be interested in (at least) the structure

$$(T, S, \text{LEFT-OF}, \text{S}, \text{NP}, \text{VP}, \text{PN}, \text{TV}).$$

Here S, NP, VP, PN, and TV are *unary* relations on T (note that S and S are distinct symbols). These relations record the information attached to each node, namely the fact that some nodes are noun phrase nodes, while others are proper name nodes, sentential nodes, and so on. LEFT-OF is a binary relation which captures the left-to-right aspect of the above picture; the fact that the NP node is to the left of the VP node might be linguistically crucial.

Similar things happen in mathematical contexts. Sometimes we will need to work with relational structures which are much richer than the simple trees (T, S) just defined, but which, perhaps in an implicit form, contain a relation with all the properties required of S. It is useful to have a general term for such structures; we will call them *tree-like*. A formal definition here would do more harm than good, but in the text we will indicate, whenever we call a structure tree-like, where this implicit tree (T, S) can be found. That is, we will say, unless it is obvious, *which* definable relation in the structure satisfies the conditions of Definition 1.7. One of the most important examples of tree-like structures is the Rabin structure, which we will meet in Section 6.3.

One often encounters the notion of a tree defined in terms of the (reflexive) transitive closure of the successor relation. Such trees we call *(reflexive and) transitive trees*, and they are dealt with in Exercises 1.1.4 and 1.1.5 $\dashv$

Example 1.8 We have already seen that labeled transition systems can be regarded as a simple model of computation. Indeed, they can be thought of as models for practically any dynamic notion: each transition takes us from an input state to an output state. But this treatment of states and transitions is rather unbalanced: it is clear that transitions are second-class citizens. For example, if we talked about LTSs using a first-order language, we couldn't name transitions using constants (they would be talked about using relation symbols) but we could have constants for states. But there is a way to treat transitions as first-class citizens: we can work with *arrow structures*.

The *objects* of an arrow structure are things that can be pictured as *arrows*. As concrete examples, the mathematically inclined reader might think of vectors, or functions or morphisms in some category; the computer scientist of programs; the linguist of the context changing potential of a grammatically well-formed piece of text or discourse; the philosopher of some agent's cognitive actions; and so on. But note well: although arrows are the prime citizens of arrow structures, this does not mean that they should always be thought of as *primitive* entities. For example, in a *two-dimensional arrow structure*, an arrow a is thought of as a *pair* (a_0, a_1) of which a_0 represents the starting point of a, and a_1 its endpoint.

Having 'defined' the elements of arrow structures to be objects graphically representable as arrows, we should now ask: what are the basic *relations* which hold between arrows? The most obvious candidate is *composition*: vector spaces have an additive structure, functions can be composed, language fragments can be concatenated, and so on. So the central relation on arrows will be a ternary *composition relation* C, where $Cabc$ says that arrow a is the outcome of composing arrow b with arrow c (or conversely, that a can be decomposed into b and c). Note that in many concrete examples, C is actually a (partial) function; for example, in the two-dimensional framework we have

$$Cabc \text{ iff } a_0 = b_0,\ a_1 = c_1 \text{ and } b_1 = c_0. \tag{1.1}$$

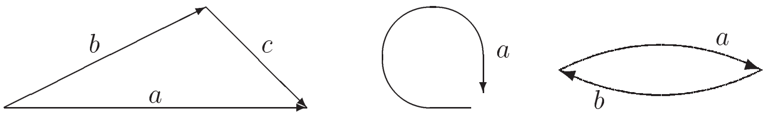
What next? Well, in all the examples listed, the composition function has a neutral element; think of the identity function or the `SKIP`-program. So, arrow structures will contain degenerate arrows, transitions that do not lead to a different state. Formally, this means that arrow structures will contain a designated subset I of *identity arrows*; in the pair-representation, I will be (a subset of) the diagonal:

$$Ia \text{ iff } a_0 = a_1. \tag{1.2}$$

Another natural relation is converse. In linguistics and cognitive science we might view this as an ‘undo’ action (perhaps we’ve made a mistake and need to recover) and in many fields of mathematics arrow-like objects have converses (vectors) or inverses (bijective functions). So we’ll also give arrow structures a binary *reverse relation* R. Again, in many cases this relation will be a partial function. For example, in the two-dimensional picture, R is given by

$$Rab \text{ iff } a_0 = b_1 \text{ and } a_1 = b_0. \tag{1.3}$$

Although there are further natural candidates for arrow relations (notably some notion of *iteration*) we’ll leave it at this. And now for the formal definition: an *arrow frame* is a quadruple $\mathfrak{F} = (W, C, R, I)$ such that C, R and I are a ternary, a binary and a unary relation on W, respectively. Pictorially, we can think of them as follows:



$Cabc$ Ia Rab

The two-dimensional arrow structure, in which the universe consists of all pairs over the set U (and the relations C, R and I are given by (1.1), (1.3) and (1.2), respectively) is called the *square over* U, notation: $\mathfrak{S}_U$. The square arrow frame over U can be pictorially represented as a full graph over U: each arrow object (a_0, a_1) in $\mathfrak{S}_U$ can be represented as a ‘real’ arrow from a_0 to a_1; the relations are as pictured above. Alternatively, square arrow frames can be represented two-dimensionally, cf. the pictures in Example 1.27. $\dashv$

Exercises for Section 1.1

1.1.1 Let (W, R) be a *quasi-order*; that is, assume that R is transitive and reflexive. Define the binary relation $\sim$ on W by putting $s \sim t$ iff Rst and Rts.

(a) Show that $\sim$ is an equivalence relation

Let $[s]$ denote the equivalence class of s under this relation, and define the following relation on the collection of equivalence classes: $[s] \leq [t]$ iff Rst.

(b) Show that this is well-defined.
(c) Show that $\leq$ is a partial order.

1.1.2 Let R be a transitive relation on a finite set W. Prove that R is well-founded iff R is irreflexive. (R is called *well-founded* if there are no infinite paths $\ldots Rs_2Rs_1Rs_0$.)

1.1.3 Let R be a binary relation on W. In Example 1.2 we defined the reflexive closure of R to be $R \cup \{(u, u) \mid u \in W\}$. But we can also give a definition analogous to those of R^+ and R^* in Definition 1.6, namely that it is the smallest reflexive relation on W that contains R:

$$R^{\mathrm{r}} = \bigcap\{R' \mid R' \text{ is a reflexive binary relation on } W \ \& \ R \subseteq R'\}.$$

Explain why this new definition (and the definitions of R^+ and R^*) are well defined. Show the equivalence of the two definitions of reflexive closure. Finally, show that R^+uv if and only if there is a sequence of elements $u = w_0, w_1, \ldots, w_n = v$ from W such that for each $i < n$ we have Rw_iw_{i+1}, and give an analogous sequence-based definition of *reflexive* transitive closure.

1.1.4 A *transitive tree* is an SPO $(T, <)$ such that (i) there is a *root* $r \in T$ satisfying $r < t$ for all $t \in T$ and (ii) for each $t \in T$, the set $\{s \in T \mid s < t\}$ of predecessors of t is finite and linearly ordered by $<$.

(a) Prove that if (T, S) is a tree then (T, S^+) is a transitive tree.
(b) Prove that $(T, <)$ is a transitive tree iff $(T, S_<)$ is a tree, where $S_<$ is the immediate successor relation given by $sS_<t$ iff $s < t$ and $s < v < t$ for no $v \in T$.
(c) Under which conditions does the converse of (a) hold?

1.1.5 Define the notion of a reflexive and transitive tree, such that if (T, S) is a tree then (T, S^*) is a reflexive and transitive tree.

1.1.6 Show that the following formulas hold on square arrow frames:

(a) $\forall xy\, (Rxy \to Ryx)$,
(b) $\forall xyz\, ((Cxyz \wedge Iz) \leftrightarrow x = y)$,
(c) $\forall xx_1x_2x_3\, (\exists y\, (Cxx_1y \wedge Cyx_2x_3) \ \leftrightarrow \ \exists z\, (Cxzx_3 \wedge Czx_1x_2))$.

1.2 Modal Languages

It's now time to meet the modal languages we will be working with. First, we introduce the *basic modal language*. We then define *modal languages of arbitrary similarity type*. Finally we examine the following extensions of the basic modal language in more detail: the *basic temporal language*, the language of *propositional dynamic logic*, and a language of *arrow logic*.

Definition 1.9 The *basic modal language* is defined using a set of *proposition letters* (or *proposition symbols* or *propositional variables*) Φ whose elements are usually denoted p, q, r, and so on, and a unary modal operator $\Diamond$ ('diamond'). The well-formed formulas ϕ of the basic modal language are given by the rule

$$\phi ::= p \mid \bot \mid \neg\phi \mid \psi \vee \phi \mid \Diamond\phi,$$

where p ranges over elements of Φ. This definition means that a formula is either a proposition letter, the propositional constant falsum ('bottom'), a negated formula, a disjunction of formulas, or a formula prefixed by a diamond.

Just as the familiar first-order existential and universal quantifiers are duals to each other (in the sense that $\forall x\, \alpha \leftrightarrow \neg\exists x\, \neg\alpha$), we have a dual operator $\Box$ ('box') for our diamond which is defined by $\Box\phi := \neg\Diamond\neg\phi$. We also make use of the classical abbreviations for conjunction, implication, bi-implication and the constant true ('top'): $\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi)$, $\phi \to \psi := \neg\phi \vee \psi$, $\phi \leftrightarrow \psi := (\phi \to \psi) \wedge (\psi \to \phi)$ and $\top := \neg\bot$. $\dashv$

Although we generally assume that the set Φ of proposition letters is a countably infinite $\{p_0, p_1, \ldots\}$, occasionally we need to make other assumptions. For instance, when we are after decidability results, it may be useful to stipulate that Φ is finite, while doing model theory or frame theory we may need uncountably infinite languages. This is why we take Φ as an explicit parameter when defining the set of modal formulas.

Example 1.10 Three readings of diamond and box have been extremely influential. First, $\Diamond\phi$ can be read as 'it is *possibly* the case that ϕ.' Under this reading, $\Box\phi$ means 'it is not possible that not ϕ,' that is, '*necessarily* ϕ,' and examples of formulas we would probably regard as correct principles include all instances of $\Box\phi \to \Diamond\phi$ ('whatever is necessary is possible') and all instances of $\phi \to \Diamond\phi$ ('whatever is, is possible'). The status of other formulas is harder to decide. Should $\phi \to \Box\Diamond\phi$ ('whatever is, is *necessarily* possible') be regarded as a general truth about necessity and possibility? Should $\Diamond\phi \to \Box\Diamond\phi$ ('whatever is possible, is necessarily possible')? Are any of these formulas linked by a modal notion of logical consequence, or are they independent claims about necessity and possibility? These are difficult (and historically important) questions. The relational semantics defined in the following section offers a simple and intuitively compelling framework in which to discuss them.

Second, in *epistemic logic* the basic modal language is used to reason about knowledge, though instead of writing $\Box\phi$ for 'the agent knows that ϕ' it is usual to write $K\phi$. Given that we are talking about knowledge (as opposed to, say, belief or rumor), it seems natural to view all instances of $K\phi \to \phi$ as true: if the agent really *knows* that ϕ, then ϕ must hold. On the other hand (assuming that the agent is not omniscient) we would regard $\phi \to K\phi$ as false. But the legitimacy of other principles is harder to judge (if an agent knows that ϕ, does she know that she knows it?). Again, a precise semantics brings clarity.

Third, in *provability logic* $\Box\phi$ is read as 'it is *provable* (in some arithmetical theory) that ϕ.' A central theme in provability logic is the search for a complete axiomatization of the provability principles that are valid for various arithmetical theories (such as Peano Arithmetic). The *Löb* formula $\Box(\Box p \to p) \to \Box p$ plays a key role here. The arithmetical ramifications of this formula lie outside the scope of the book, but in Chapters 3 and 4 we will explore its modal content. $\dashv$

That's the basic modal language. Let's now generalize it. There are two obvious ways to do so. First, there seems no good reason to restrict ourselves to languages with only one diamond. Second, there seems no good reason to restrict ourselves to modalities that take only a single formula as argument. Thus the general modal languages we will now define may contain many modalities, of arbitrary arities.

Definition 1.11 A *modal similarity type* is a pair $\tau = (O, \rho)$ where O is a non-empty set, and ρ is a function $O \to \mathbb{N}$. The elements of O are called *modal operators*; we use $\triangle$ ('*triangle*'), $\triangle_0, \triangle_1, \ldots$ to denote elements of O. The function ρ assigns to each operator $\triangle \in O$ a finite *arity*, indicating the number of arguments $\triangle$ can be applied to.

In line with Definition 1.9, we often refer to *unary* triangles as *diamonds*, and denote them by $\Diamond_a$ or $\langle a \rangle$, where a is taken from some index set. We often assume that the arity of operators is known, and do not distinguish between τ and O. $\dashv$

Definition 1.12 A *modal language* $ML(\tau, \Phi)$ is built up using a modal similarity type $\tau = (O, \rho)$ and a set of proposition letters Φ. The set $Form(\tau, \Phi)$ of *modal formulas* over τ and Φ is given by the rule

$$\phi := p \mid \bot \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \triangle(\phi_1, \ldots, \phi_{\rho(\triangle)}),$$

where p ranges over elements of Φ. $\dashv$

The similarity type of the basic modal language is called τ_0. In the sequel we sometimes state results for modal languages of arbitrary similarity types, give the proof for similarity types with diamonds only, and leave the general case as an exercise. For binary modal operators, we often use infix notation; that is, we usually write $\phi\triangle\psi$ instead of $\triangle(\phi, \psi)$. One other thing: note that our definition permits *nullary modalities* (or *modal constants*), triangles that take no arguments at all. Such modalities can be useful — we will see a natural example when we discuss arrow logic — but they play a relatively minor role in this book. Syntactically (and indeed, semantically) they are rather like propositional variables; in fact, they are best thought of as propositional *constants*.

Definition 1.13 We now define dual operators for non-nullary triangles. For each $\triangle \in O$ the *dual* ∇ of $\triangle$ is defined as $\nabla(\phi_1, \ldots, \phi_n) := \neg\triangle(\neg\phi_1, \ldots, \neg\phi_n)$. The dual of a triangle of arity at least 2 is called a *nabla*. As in the basic modal language, the dual of a diamond is called a *box*, and is written $\Box_a$ or $[a]$. $\dashv$

Three extensions of the basic modal language deserve special attention. Two of these, the *basic temporal language* and the language of *propositional dynamic logic* will be frequently used in subsequent chapters. The third is a simple language of *arrow logic*; it will provide us with a natural example of a binary modality.

Example 1.14 (The Basic Temporal Language) The basic temporal language is built using a set of unary operators $O = \{\langle F \rangle, \langle P \rangle\}$. The intended interpretation of a formula $\langle F\rangle\phi$ is 'ϕ will be true at some *F*uture time,' and the intended interpretation of $\langle P\rangle\phi$ is 'ϕ was true at some *P*ast time.' This language is called the *basic temporal language*, and it is the core language underlying a branch of modal logic called *temporal logic*. It is traditional to write $\langle F\rangle$ as F and $\langle P\rangle$ as P, and their duals are written as G and H, respectively. (The mnemonics here are: 'it is always *G*oing to be the case' and 'it always *H*as been the case.')

We can express many interesting assertions about time with this language. For example, $P\phi \to GP\phi$, says 'whatever has happened will always have happened,' and this seems a plausible candidate for a general truth about time. On the other hand, if we insist that $F\phi \to FF\phi$ must always be true, it shows that we are thinking of time as *dense*: between any two instants there is always a third. And if we insist that $GFp \to FGp$ (the *McKinsey formula*) is true, for all propositional symbols p, we are insisting that atomic information true somewhere in the future eventually settles down to being always true. (We might think of this as reflecting a 'thermodynamic' view of information distribution.)

One final remark: computer scientists will have noticed that the binary until modality is conspicuous by its absence. As we will see in the following chapter, the basic temporal language is *not* strong enough to express until. We examine a language containing the until operator in Section 7.2. $\dashv$

Example 1.15 (Propositional Dynamic Logic) Another important branch of modal logic, again involving only unary modalities, is *propositional dynamic logic*. PDL, the language of propositional dynamic logic, has an infinite collection of diamonds. Each of these diamonds has the form $\langle\pi\rangle$, where π denotes a (non-deterministic) *program*. The intended interpretation of $\langle\pi\rangle\phi$ is 'some terminating execution of π from the present state leads to a state bearing the information ϕ.' The dual assertion $[\pi]\phi$ states that 'every execution of π from the present state leads to a state bearing the information ϕ.'

So far, there's nothing really new — but a simple idea is going to ensure that PDL is highly expressive: we will make the inductive structure of the programs explicit in PDL's syntax. Complex programs are built out of basic programs using some repertoire of program constructors. By using diamonds which reflect this structure, we obtain a powerful and flexible language.

Let us examine the core language of PDL. Suppose we have fixed some set of basic programs a, b, c, and so on (thus we have basic modalities $\langle a\rangle$, $\langle b\rangle$, $\langle c\rangle$, ... at our disposal). Then we are allowed to define complex programs π (and hence, modal operators $\langle\pi\rangle$) over this base as follows:

(choice) if π_1 and π_2 are programs, then so is $\pi_1 \cup \pi_2$.
The program $\pi_1 \cup \pi_2$ (non-deterministically) executes π_1 or π_2.

(composition) if π_1 and π_2 are programs, then so is $\pi_1 ; \pi_2$.
This program first executes π_1 and then π_2.

(iteration) if π is a program, then so is π^*.
π^* is a program that executes π a finite (possibly zero) number of times.

For the collection of diamonds this means that if $\langle \pi_1 \rangle$ and $\langle \pi_2 \rangle$ are modal operators, then so are $\langle \pi_1 \cup \pi_2 \rangle$, $\langle \pi_1 ; \pi_2 \rangle$ and $\langle \pi_1^* \rangle$. This notation makes it straightforward to describe properties of program execution. Here is a fairly straightforward example. The formula $\langle \pi^* \rangle \phi \leftrightarrow \phi \vee \langle \pi ; \pi^* \rangle \phi$ says that a state bearing the information ϕ can be reached by executing π a finite number of times if and only if either we already have the information ϕ in the current state, or we can execute π once and then find a state bearing the information ϕ after finitely many more iterations of π. Here's a far more demanding example:

$$[\pi^*](\phi \to [\pi]\phi) \to (\phi \to [\pi^*]\phi).$$

This is *Segerberg's axiom* (or the *induction axiom*) and the reader should try working out what exactly it is that this formula says. We discuss this formula further in Chapter 3, cf. Example 3.10.

If we confine ourselves to these three constructors (and in this book for the most part we do) we are working with a version of PDL called *regular* PDL. (This is because the three constructors are the ones used in Kleene's well-known analysis of regular programs.) However, a wide range of other constructors have been studied. Here are two:

(intersection) if π_1 and π_2 are programs, then so is $\pi_1 \cap \pi_2$.
The intended meaning of $\pi_1 \cap \pi_2$ is: execute both π_1 and π_2, in parallel.

(test) if ϕ is a formula, then $\phi?$ is a program.
This program tests whether ϕ holds, and if so, continues; if not, it fails.

To flesh this out a little, the intended reading of $\langle \pi_1 \cap \pi_2 \rangle \phi$ is that if we execute both π_1 and π_2 in the present state, then there is at least one state reachable by both programs which bears the information ϕ. This is a natural constructor for a variety of purposes, and we will make use of it in Section 6.5.

The key point to note about the test constructor is its unusual syntax: it allows us to make a modality out of a formula. Intuitively, this modality accesses the *current* state if the current state satisfies ϕ. On its own such a constructor is uninteresting ($\langle \phi? \rangle \psi$ simply means $\phi \wedge \psi$). However, when other constructors are present, it can be used to build interesting programs. For example, $(p? ; a) \cup (\neg p? ; b)$ is '`if` p `then` a `else` b.'

Nothing prevents us from viewing the basic programs as *deterministic*, and we will discuss a fragment of deterministic PDL (DPDL) in Section 6.5 $\dashv$

Example 1.16 (An Arrow Language) A similarity type with modal operators other than diamonds, is the type $\tau_{\rightarrow}$ of *arrow logic*. The language of arrow logic is designed to talk about the objects in arrow structures (entities which can be pictured as arrows). The well-formed formulas ϕ of the arrow language are given by the rule

$$\phi := p \mid \bot \mid \neg\phi \mid \phi \vee \psi \mid \phi \circ \psi \mid \otimes\phi \mid 1'.$$

That is, 1' ('identity') is a nullary modality (a modal constant), the 'converse' operator $\otimes$ is a diamond, and the 'composition' operator $\circ$ is a dyadic operator. Possible readings of these operators are:

1'	identity	'skip'
$\otimes\phi$	converse	'ϕ conversely'
$\phi \circ \psi$	composition	'first ϕ, then ψ'. $\dashv$

Example 1.17 (Feature Logic and Description Logic) As we mentioned in the Preface, researchers developing formalisms for describing graphs have sometimes (without intending to) come up with notational variants of modal logic. For example, computational linguists use *Attribute-Value Matrices* (AVMs) for describing *feature structures* (directed acyclic graphs that encode linguistic information). Here's a fairly typical AVM:

$$\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & \textit{1st} \\ \text{NUMBER} & \textit{plural} \end{bmatrix} \\ \text{CASE} & \textit{dative} \end{bmatrix}$$

But this is just a two dimensional notation for the following modal formula

$$\langle\text{AGREEMENT}\rangle(\langle\text{PERSON}\rangle \textit{1st} \wedge \langle\text{NUMBER}\rangle \textit{plural}) \quad \wedge \\ \langle\text{CASE}\rangle \textit{dative}$$

Similarly, researchers in AI needing a notation for describing and reasoning about ontologies developed *description logic*. For example, the concept of 'being a hired killer for the mob' is true of any individual who is a killer and is employed by a gangster. In description logic we can define this concept as follows:

$$\texttt{killer} \sqcap \exists\texttt{employer.gangster}$$

But this is simply the following modal formula lightly disguised:

$$\texttt{killer} \wedge \langle\texttt{employer}\rangle\texttt{gangster}$$

It turns out that the links between modal logic on the one hand, and feature and description logic on the other, are far more interesting than these rather simple examples might suggest. A modal perspective on feature or description logic capable of accounting for other important aspects of these systems (such as the ability to talk about re-entrancy in feature structures, or to perform ABox reasoning in description logic) must make use of the kinds of extended modal logics discussed in Chapter 7 (in particular, logics containing the global modality, and hybrid logics). Furthermore, some versions of feature and description logic make use of ideas from PDL, and description logic makes heavy use of *counting modalities* (which say such things as 'at most 3 transitions lead to a ϕ state'). $\dashv$

Substitution

Throughout this book we'll be working with the syntactic notion of one formula being a substitution instance of another. In order to define this notion we first introduce the concept of a substitution as a function mapping proposition letters to variables.

Definition 1.18 Suppose we're working a modal similarity type τ and a set Φ of proposition letters. A *substitution* is a map $\sigma : \Phi \to Form(\tau, \Phi)$.

Now such a substitution σ induces a map $(\cdot)^\sigma : Form(\tau, \Phi) \to Form(\tau, \Phi)$ which we can recursively define as follows:

$$\begin{aligned}
\bot^\sigma &= \bot \\
p^\sigma &= \sigma(p) \\
(\neg\psi)^\sigma &= \neg\psi^\sigma \\
(\psi \vee \theta)^\sigma &= \psi^\sigma \vee \theta^\sigma \\
(\triangle(\psi_1, \ldots, \psi_n))^\sigma &= \triangle(\psi_1^\sigma, \ldots, \psi_n^\sigma).
\end{aligned}$$

This definition spells out exactly what is meant by carrying out *uniform substitution*. Finally, we say that χ is a *substitution instance* of ψ if there is some substitution τ such that $\psi^\tau = \chi$. $\dashv$

To give an example, if σ is the substitution that maps p to $p \wedge \Box q$, q to $\Diamond\Diamond q \vee r$ and leaves all other proposition letters untouched, then we have

$$(p \wedge q \wedge r)^\sigma \;=\; ((p \wedge \Box q) \wedge (\Diamond\Diamond q \vee r) \wedge r).$$

Exercises for Section 1.2

1.2.1 Using $K\phi$ to mean 'the agent knows that ϕ' and $M\phi$ to mean 'it is consistent with what the agent knows that ϕ,' represent the following statements.

(a) If ϕ is true, then it is consistent with what the agent knows that she knows that ϕ.
(b) If it is consistent with what the agent knows that ϕ, and it is consistent with what the agent knows that ψ, then it is consistent with what the agent knows that $\phi \wedge \psi$.
(c) If the agent knows that ϕ, then it is consistent with what the agent knows that ϕ.

(d) If it is consistent with what the agent knows that it is consistent with what the agent knows that ϕ, then it is consistent with what the agent knows that ϕ.

Which of these seem plausible principles concerning knowledge and consistency?

1.2.2 Suppose $\Diamond\phi$ is interpreted as 'ϕ is permissible'; how should $\Box\phi$ be understood? List formulas which seem plausible under this interpretation. Should the Löb formula $\Box(\Box p \to p) \to \Box p$ be on your list? Why?

1.2.3 Explain how the program constructs '`while` ϕ `do` π' and '`repeat` π `until` ϕ' can be expressed in PDL.

1.2.4 Consider the following arrow formulas. Do you think they should be always true?

$$\begin{array}{rcl} 1' \circ p & \leftrightarrow & p, \\ \otimes(p \circ q) & \leftrightarrow & \otimes q \circ \otimes p, \\ p \circ (q \circ r) & \leftrightarrow & (p \circ q) \circ r. \end{array}$$

1.2.5 Show that 'being-a-substitution-instance-of' is a transitive concept. That is, show that if χ is a substitution instance of ψ, and ψ is a substitution instance of ϕ, then χ is a substitution instance of ϕ.

1.3 Models and Frames

Although our discussion has contained many semantically suggestive phrases such as 'true' and 'intended interpretation', as yet we have given them no mathematical content. The purpose of this (key) section is to put that right. We do so by interpreting our modal languages in relational structures. In fact, by the end of the section we will have done this in two distinct ways: at the level of *models* and at the level of *frames*. Both levels are important, though in different ways. The level of models is important because this is where the fundamental notion of *satisfaction* (or *truth*) is defined. The level of frames is important because it supports the key logical notion of *validity*.

Models and satisfaction

We start by defining frames, models, and the satisfaction relation for the basic modal language.

Definition 1.19 A *frame* for the basic modal language is a pair $\mathfrak{F} = (W, R)$ such that

(i) W is a non-empty set.
(ii) R is a binary relation on W.

That is, a frame for the basic modal language is simply a relational structure bearing a single binary relation. We remind the reader that we refer to the elements of W by many different names (see Definition 1.1).

A *model* for the basic modal language is a pair $\mathfrak{M} = (\mathfrak{F}, V)$, where $\mathfrak{F}$ is a frame for the basic modal language, and V is a function assigning to each proposition letter p in Φ a subset $V(p)$ of W. Informally we think of $V(p)$ as the set of points in our model where p is true. The function V is called a *valuation*. Given a model $\mathfrak{M} = (\mathfrak{F}, V)$, we say that $\mathfrak{M}$ is *based on* the frame $\mathfrak{F}$, or that $\mathfrak{F}$ is the frame *underlying* $\mathfrak{M}$. $\dashv$

Note that models for the basic modal language can be viewed as relational structures in a natural way, namely as structures of the form:

$$(W, R, V(p), V(q), V(r), \ldots).$$

That is, a model is a relational structure consisting of a domain, a single binary relation R, and the unary relations given to us by V. Thus, viewed from a purely structural perspective, a frame $\mathfrak{F}$ and a model $\mathfrak{M}$ based on $\mathfrak{F}$, are simply two relational models based on the same universe; indeed, a model is simply a frame enriched by a collection of unary relations.

But in spite of their mathematical kinship, frames and models are *used* very differently. Frames are essentially mathematical pictures of ontologies that we find interesting. For example, we may view time as a collection of points ordered by a strict partial order, or feel that a correct analysis of knowledge requires that we postulate the existence of situations linked by a relation of 'being an epistemic alternative to.' In short, we use the level of frames to make our fundamental assumptions mathematically precise.

The unary relations provided by valuations, on the other hand, are there to dress our frames with contingent information. Is it raining on Tuesday or not? Is the system write-enabled at time t_6? Is a situation where Janet does not love him an epistemic alternative for John? Such information is important, and we certainly need to be able to work with it — nonetheless, statements only deserve the description 'logical' if they are *invariant* under changes of contingent information. Because we have drawn a distinction between the fundamental information given by frames, and the additional descriptive content provided by models, it will be straightforward to define a modally reasonable notion of validity.

But this is jumping ahead. First we must learn how to interpret the basic modal language in models. This we do by means of the following satisfaction definition.

Definition 1.20 Suppose w is a state 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 state w as follows:

$$\begin{array}{rll}
\mathfrak{M}, w \Vdash p & \text{iff} & w \in V(p), \quad \text{where } p \in \Phi \\
\mathfrak{M}, w \Vdash \bot & & \text{never} \\
\mathfrak{M}, w \Vdash \neg\phi & \text{iff} & \text{not } \mathfrak{M}, w \Vdash \phi \\
\mathfrak{M}, w \Vdash \phi \vee \psi & \text{iff} & \mathfrak{M}, w \Vdash \phi \text{ or } \mathfrak{M}, w \Vdash \psi \\
\mathfrak{M}, w \Vdash \Diamond\phi & \text{iff} & \text{for some } v \in W \text{ with } Rwv \text{ we have } \mathfrak{M}, v \Vdash \phi.
\end{array} \quad (1.4)$$

It follows from this definition that $\mathfrak{M}, w \Vdash \Box\phi$ if and only if for all $v \in W$ such that Rwv, we have $\mathfrak{M}, v \Vdash \phi$. Finally, we say that a *set* Σ of formulas is true at a state w of a model $\mathfrak{M}$, notation: $\mathfrak{M}, w \Vdash \Sigma$, if all members of Σ are true at w. $\dashv$

Note that this notion of satisfaction is intrinsically *internal* and *local*. We evaluate formulas *inside* models, at some particular state w (the *current state*). Moreover, $\Diamond$ works locally: the final clause (1.4) treats $\Diamond\phi$ as an instruction to scan states in search of one where ϕ is satisfied. Crucially, only states R-accessible from the current one can be scanned by our operators. Much of the characteristic flavor of modal logic springs from the perspective on relational structures embodied in the satisfaction definition.

If $\mathfrak{M}$ does not satisfy ϕ at w we often write $\mathfrak{M}, w \nVdash \phi$, and say that ϕ is *false* or *refuted* at w. When $\mathfrak{M}$ is clear from the context, we write $w \Vdash \phi$ for $\mathfrak{M}, w \Vdash \phi$ and $w \nVdash \phi$ for $\mathfrak{M}, w \nVdash \phi$. It is convenient to extend the valuation V from proposition letters to arbitrary formulas so that $V(\phi)$ always denotes the set of states at which ϕ is true:

$$V(\phi) \;:=\; \{w \mid \mathfrak{M}, w \Vdash \phi\}.$$

Definition 1.21 A formula ϕ is *globally* or *universally true* in a model $\mathfrak{M}$ (notation: $\mathfrak{M} \Vdash \phi$) if it is satisfied at all points in $\mathfrak{M}$ (that is, if $\mathfrak{M}, w \Vdash \phi$, for all $w \in W$). A formula ϕ is *satisfiable* in a model $\mathfrak{M}$ if there is *some* state in $\mathfrak{M}$ at which ϕ is true; a formula is *falsifiable* or *refutable* in a model if its negation is satisfiable.

A *set* Σ of formulas is globally true (satisfiable, respectively) in a model $\mathfrak{M}$ if $\mathfrak{M}, w \Vdash \Sigma$ for all states w in $\mathfrak{M}$ (some state w in $\mathfrak{M}$, respectively). $\dashv$

Example 1.22 (i) Consider the frame $\mathfrak{F} = (\{w_1, w_2, w_3, w_4, w_5\}, R)$, where Rw_iw_j iff $j = i + 1$:

$$\begin{tikzcd} \underset{w_1}{\bullet} \arrow[r] & \underset{w_2}{\bullet} \arrow[r] & \underset{w_3}{\bullet} \arrow[r] & \underset{w_4}{\bullet} \arrow[r] & \underset{w_5}{\bullet} \end{tikzcd}$$

If we choose a valuation V on $\mathfrak{F}$ such that $V(p) = \{w_2, w_3\}$, $V(q) = \{w_1, w_2, w_3, w_4, w_5\}$, and $V(r) = \varnothing$, then in the model $\mathfrak{M} = (\mathfrak{F}, V)$ we have that $\mathfrak{M}, w_1 \Vdash$

$\Diamond\Box p$, $\mathfrak{M}, w_1 \not\Vdash \Diamond\Box p \to p$, $\mathfrak{M}, w_2 \Vdash \Diamond(p \wedge \neg r)$, and $\mathfrak{M}, w_1 \Vdash q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$.

Furthermore, $\mathfrak{M} \Vdash \Box q$. Now, it is clear that $\Box q$ is true at w_1, w_2, w_3 and w_4, but why is it true at w_5? Well, as w_5 has no successors at all (we often call such points '*dead ends*' or '*blind states*') it is vacuously true that q is true at all R-successors of w_5. Indeed, any 'boxed' formula $\Box\phi$ is true at any dead end in any model.

(ii) As a second example, let $\mathfrak{F}$ be the SPO given in Figure 1.1, where $W = \{1$, 2, 3, 4, 6, 8, 12, 24$\}$ and Rxy means 'x and y are different, and y can be divided by x.' Choose a valuation V on this frame such that $V(p) = \{4, 8, 12, 24\}$, and $V(q) = \{6\}$, and let $\mathfrak{M} = (\mathfrak{F}, V)$. Then $\mathfrak{M}, 4 \Vdash \Box p$, $\mathfrak{M}, 6 \Vdash \Box p$, $\mathfrak{M}, 2 \not\Vdash \Box p$, and $\mathfrak{M}, 2 \Vdash \Diamond(q \wedge \Box p) \wedge \Diamond(\neg q \wedge \Box p)$.

(iii) Whereas a diamond $\Diamond$ corresponds to making a single R-step in a model, stacking diamonds one in front of the other corresponds to making a sequence of R-steps through the model. The following defined operators will sometimes be useful: we write $\Diamond^n\phi$ for ϕ preceded by n occurrences of $\Diamond$, and $\Box^n\phi$ for ϕ preceded by n occurrences of $\Box$. If we like, we can associate each of these defined operators with its own accessibility relation. We do so inductively: R^0xy is defined to hold if $x = y$, and $R^{n+1}xy$ is defined to hold if $\exists z\,(Rxz \wedge R^n zy)$. Under this definition, for any model $\mathfrak{M}$ and state w in $\mathfrak{M}$ we have $\mathfrak{M}, w \Vdash \Diamond^n\phi$ iff there exists a v such that $R^n wv$ and $\mathfrak{M}, v \Vdash \phi$.

(iv) The use of the word 'world' (or 'possible world') for the entities in W derives from the reading of the basic modal language in which $\Diamond\phi$ is taken to mean '*possibly* ϕ,' and $\Box\phi$ to mean '*necessarily* ϕ.' Given this reading, the machinery of frames, models, and satisfaction which we have defined is essentially an attempt to capture mathematically the view (often attributed to Leibniz) that *necessity* means *truth in all possible worlds*, and that *possibility* means *truth in some possible world*.

The satisfaction definition stipulates that $\Diamond$ and $\Box$ check for truth not at *all* possible worlds (that is, at all elements of W) but only at R-accessible possible worlds. At first sight this may seem a weakness of the satisfaction definition — but in fact, it's its greatest source of strength. The point is this: varying R is a mechanism which gives us a firm mathematical grip on the pre-theoretical notion of access between possible worlds. For example, by stipulating that $R = W \times W$ we can allow all worlds access to each other; this corresponds to the Leibnizian idea in its purest form. Going to the other extreme, we might stipulate that *no* world has access to any other. Between these extremes there is a wide range of options to explore. Should interworld access be reflexive? Should it be transitive? What impact do these choices have on the notions of necessity and possibility? For example, if we demand symmetry, does this justify certain principles, or rule others out?

(v) Recall from Example 1.10 that in epistemic logic $\Box$ is written as K and $K\phi$ is interpreted as 'the agent knows that ϕ.' Under this interpretation, the intuitive reading for the semantic clause governing K is: the agent knows ϕ in a situation

w (that is, $w \Vdash K\phi$) iff ϕ is true in all situations v that are compatible with her knowledge (that is, if $v \Vdash \phi$ for all v such that Rwv). Thus, under this interpretation, W is to be thought of as a collection of situations, R is a relation which models the idea of one situation being epistemically accessible from another, and V governs the distribution of primitive information across situations. $\dashv$

We now define frames, models and satisfaction for modal languages of arbitrary similarity type.

Definition 1.23 Let τ be a modal similarity type. A τ-*frame* is a tuple $\mathfrak{F}$ consisting of the following ingredients:

(i) a non-empty set W,
(ii) for each $n \geq 0$, and each n-ary modal operator $\triangle$ in the similarity type τ, an $(n+1)$-ary relation $R_\triangle$.

So, again, frames are simply relational structures. If τ contains just a finite number of modal operators $\triangle_1, \ldots, \triangle_n$, we write $\mathfrak{F} = (W, R_{\triangle_1}, \ldots, R_{\triangle_n})$; otherwise we write $\mathfrak{F} = (W, R_\triangle)_{\triangle \in \tau}$ or $\mathfrak{F} = (W, \{R_\triangle \mid \triangle \in \tau\})$. We turn such a frame into a model in exactly the same way we did for the basic modal language: by adding a valuation. That is, a τ-*model* is a pair $\mathfrak{M} = (\mathfrak{F}, V)$ where $\mathfrak{F}$ is a τ-frame, and V is a valuation with domain Φ and range $\mathcal{P}(W)$, where W is the universe of $\mathfrak{F}$.

The notion of a formula ϕ being *satisfied* (or *true*) *at a state* w in a model $\mathfrak{M} = (W, \{R_\triangle \mid \triangle \in \tau\}, V)$ (notation: $\mathfrak{M}, w \Vdash \phi$) is defined inductively. The clauses for the atomic and Boolean cases are the same as for the basic modal language (see Definition 1.20). As for the modal case, when $\rho(\triangle) > 0$ we define

$$\begin{array}{lll} \mathfrak{M}, w \Vdash \triangle(\phi_1, \ldots, \phi_n) & \text{iff} & \text{for some } v_1, \ldots, v_n \in W \text{ with } R_\triangle w v_1 \ldots v_n \\ & & \text{we have, for each } i,\ \mathfrak{M}, v_i \Vdash \phi_i. \end{array}$$

This is an obvious generalization of the way $\Diamond$ is handled in the basic modal language. Before going any further, the reader should formulate the satisfaction clause for $\nabla(\phi_1, \ldots, \phi_n)$.

On the other hand, when $\rho(\triangle) = 0$ (that is, when $\triangle$ is a nullary modality) then $R_\triangle$ is a unary relation and we define

$$\mathfrak{M}, w \Vdash \triangle \quad \text{iff} \quad w \in R_\triangle.$$

That is, unlike other modalities, nullary modalities do not access other states. In fact, their semantics is identical to that of the propositional variables, save that the unary relations used to interpret them are *not* given by the valuation — rather, they are part of the underlying *frame*.

As before, we often write $w \Vdash \phi$ for $\mathfrak{M}, w \Vdash \phi$ where $\mathfrak{M}$ is clear from the context. The concept of *global truth* (or *universal truth*) in a model is defined as for the basic modal language: it simply means *truth at all states in the model*. And, as before, we sometimes extend the valuation V supplied by $\mathfrak{M}$ to arbitrary formulas. $\dashv$

Example 1.24 (i) Let τ be a similarity type with three unary operators $\langle a \rangle$, $\langle b \rangle$, and $\langle c \rangle$. Then a τ-frame has three binary relations R_a, R_b, and R_c (that is, it is a labeled transition system with three labels). To give an example, let W, R_a, R_b and R_c be as in Figure 1.2, and consider the formula $\langle a \rangle p \to \langle b \rangle p$. Informally, this formula is true at a state, if it has an R_a-successor satisfying p only if it has an R_b-successor satisfying p. Let V be a valuation with $V(p) = \{w_2\}$. Then the model $\mathfrak{M} = (W, R_a, R_b, R_c, V)$ has $\mathfrak{M}, w_1 \not\Vdash \langle a \rangle p \to \langle b \rangle p$.

(ii) Let τ be a similarity type with a binary modal operator $\triangle$ and a ternary operator $\bigcirc$. Frames for this τ contain a ternary relation $R_\triangle$ and a 4-ary relation $S_\bigcirc$. As an example, let $W = \{u, v, w, s\}$, $R_\triangle = \{(u, v, w)\}$, and $S_\bigcirc = \{(u, v, w, s)\}$ as in Figure 1.6, and consider a valuation V on this frame with $V(p_0) = \{v\}$, $V(p_1) = \{w\}$ and $V(p_2) = \{s\}$. Now, let ϕ be the formula

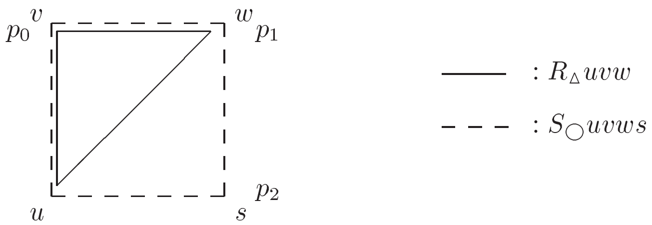


Fig. 1.6. A simple frame

$\triangle(p_0, p_1) \to \bigcirc(p_0, p_1, p_2)$. An informal reading of ϕ is 'any triangle of which the evaluation point is a vertex, and which has p_0 and p_1 true at the other two vertices, can be expanded to a rectangle with a fourth point at which p_2 is true.' The reader should be able to verify that ϕ is true at u, and indeed at all other points, and hence that it is globally true in the model. $\dashv$

Example 1.25 (Bidirectional Frames and Models) Recall from Example 1.14 that the basic temporal language has two unary operators F and P. Thus, according to Definition 1.23, models for this language consist of a set bearing two binary relations, R_F (the into-the-future relation) and R_P (the into-the-past relation), which are used to interpret F and P respectively. However, given the intended reading of the operators, most such models are inappropriate: clearly we ought to insist on working with models based on frames in which R_P is the *converse* of R_F (that is, frames in which $\forall xy\,(R_F xy \leftrightarrow R_P yx)$).

Let us denote the converse of a relation R by $R^{\breve{}}$. We will call a frame of the form $(T, R, R^{\smile})$ a *bidirectional frame*, and a model built over such a frame a *bidirectional model*. From now on, we will only interpret the basic temporal language in bidirectional models. That is, if $\mathfrak{M} = (T, R, R^{\smile}, V)$ is a bidirectional model then:

$$\begin{array}{lll} \mathfrak{M}, t \Vdash F\phi & \text{iff} & \exists s\,(Rts \ \wedge\ \mathfrak{M}, s \Vdash \phi) \\ \mathfrak{M}, t \Vdash P\phi & \text{iff} & \exists s\,(R^{\smile}ts \ \wedge\ \mathfrak{M}, s \Vdash \phi). \end{array}$$

But of course, once we've made this restriction, we don't need to mention $R^{\smile}$ explicitly any more: once R has been fixed, its converse is fixed too. That is, we are free to interpret the basic temporal languages on frames (T, R) for the basic modal language using the clauses

$$\begin{array}{lll} \mathfrak{M}, t \Vdash F\phi & \text{iff} & \exists s\,(Rts \ \wedge\ \mathfrak{M}, s \Vdash \phi) \\ \mathfrak{M}, t \Vdash P\phi & \text{iff} & \exists s\,(Rst \ \wedge\ \mathfrak{M}, s \Vdash \phi). \end{array}$$

These clauses clearly capture a crucial part of the intended semantics: F looks forward along R, and P looks backwards along R. Of course, our models will only start looking genuinely *temporal* when we insist that R has further properties (notably transitivity, to capture the flow of time), but at least we have pinned down the fundamental interaction between the two modalities. $\dashv$

Example 1.26 (Regular Frames and Models) As explained in Example 1.15, the language of PDL has an infinite collection of diamonds, each indexed by a program π built from basic programs using the constructors $\cup$, ;, and $*$. Now, according to Definition 1.23, a model for this language has the form

$$(W, \{R_\pi \mid \pi \text{ is a program }\}, V).$$

That is, a model is a labeled transition system together with a valuation. However, given our reading of the PDL operators, most of these models are uninteresting. As with the basic temporal language, we must insist on working with a class of models that does justice to our intentions.

Now, there is no problem with the interpretation of the basic programs: any binary relation can be regarded as a transition relation for a non-deterministic program. Of course, if we were particularly interested in *deterministic* programs we would insist that each basic program be interpreted by a partial function, but let us ignore this possibility and turn to the key question: which relations should interpret the structured modalities? Given our readings of $\cup$, ; and *, as choice, composition, and iteration, it is clear that we are only interested in relations constructed using the following inductive clauses:

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

These inductive clauses completely determine how each modality should be interpreted. Once the interpretation of the basic programs has been fixed, the relation corresponding to each complex program is fixed too. This leads to the following definition.

Suppose we have fixed a set of basic programs. Let Π be the smallest set of programs containing the basic programs and all programs constructed over them using the regular constructors $\cup$, ; and *. Then a *regular frame for Π* is a labeled transition system $(W, \{R_\pi \mid \pi \in \Pi\})$ such that R_a is an arbitrary binary relation for each basic program a, and for all complex programs π, R_π is the binary relation inductively constructed in accordance with the previous clauses. A *regular model* for Π is a model built over a regular frame; that is, a regular model is regular frame together with a valuation. When working with the language of PDL over the programs in Π, we will only be interested in regular models for Π, for these are the models that capture the intended interpretation.

What about the $\cap$ and ? constructors? Clearly the intended reading of $\cap$ demands that $R_{\pi_1 \cap \pi_2} = R_{\pi_1} \cap R_{\pi_2}$. As for ?, it is clear that we want the following definition:

$$R_{\phi?} \quad = \quad \{(x, y) \mid x = y \text{ and } y \Vdash \phi\}.$$

This is indeed the clause we want, but note that it is rather different from the others: it is not a *frame* condition. Rather, in order to determine the relation $R_{\phi?}$, we need information about the *truth* of the formula ϕ, and this can only be provided at the level of *models*. $\dashv$

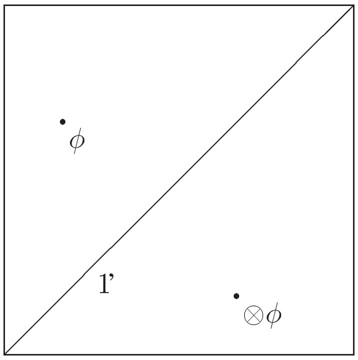
Example 1.27 (Arrow Models) Arrow frames were defined in Example 1.8 and the arrow language in Example 1.16. Given these definitions, it is clear how the language of arrow logic should be interpreted. First, an *arrow model* is a structure $\mathfrak{M} = (\mathfrak{F}, V)$ such that $\mathfrak{F} = (W, C, R, I)$ is an arrow frame and V is a valuation. Then:

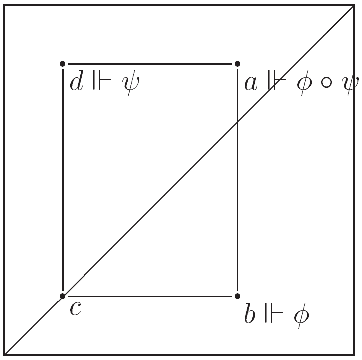
$$\begin{array}{rll} \mathfrak{M}, a \Vdash 1' & \text{iff} & Ia, \\ \mathfrak{M}, a \Vdash \otimes\phi & \text{iff} & \mathfrak{M}, b \Vdash \phi \text{ for some } b \text{ with } Rab, \\ \mathfrak{M}, a \Vdash \phi \circ \psi & \text{iff} & \mathfrak{M}, b \Vdash \phi \text{ and } \mathfrak{M}, c \Vdash \psi \text{ for some } b \text{ and } c \text{ with } Cabc. \end{array}$$

When $\mathfrak{F}$ is a *square* frame $\mathfrak{S}_U$ (as defined in Example 1.8), this works out as follows. V now maps propositional variables to sets of *pairs* over U; that is, to binary relations. The truth definition can be rephrased as follows:

$$\begin{array}{rll} \mathfrak{M}, (a_0, a_1) \Vdash 1' & \text{iff} & a_0 = a_1, \\ \mathfrak{M}, (a_0, a_1) \Vdash \otimes\phi & \text{iff} & \mathfrak{M}, (a_1, a_0) \Vdash \phi \\ \mathfrak{M}, (a_0, a_1) \Vdash \phi \circ \psi & \text{iff} & \mathfrak{M}, (a_0, u) \Vdash \phi \text{ and } \mathfrak{M}, (u, a_1) \Vdash \psi \text{ for some } u \in U. \end{array}$$

Such situations can be represented pictorially in two ways. First, one could draw the graph-like structures as given in Example 1.8. Alternatively, one could draw a square model two-dimensionally, as in the picture below. It will be obvious that the modal constant 1' holds precisely at the *diagonal points* and that $\otimes\phi$ is true at a point iff ϕ holds at its *mirror image* with respect to the diagonal. The formula $\phi \circ \psi$ holds at a point a iff we can draw a rectangle $abcd$ such that: b lies on the vertical line through a, d lies on the vertical line through a; and c lies on the diagonal.





⊣

Frames and validity

It is time to define one of the key concepts in modal logic. So far we have been viewing modal languages as tools for talking about models. But models are composite entities consisting of a frame (our underlying ontology) and contingent information (the valuation). We often want to ignore the effects of the valuation and get a grip on the more fundamental level of frames. The concept of *validity* lets us do this. A formula is valid on a frame if it is true at every state in every model that can be built over the frame. In effect, this concept interprets modal formulas on frames by abstracting away from the effects of particular valuations.

Definition 1.28 A formula ϕ is *valid at a state w in a frame $\mathfrak{F}$* (notation: $\mathfrak{F}, w \Vdash \phi$) if ϕ is true at w in every model $(\mathfrak{F}, V)$ based on $\mathfrak{F}$; ϕ is *valid in a frame $\mathfrak{F}$* (notation: $\mathfrak{F} \Vdash \phi$) if it is valid at every state in $\mathfrak{F}$. A formula ϕ is *valid on a class of frames* F (notation: $\mathsf{F} \Vdash \phi$) if it is valid on every frame $\mathfrak{F}$ in F; and it is *valid* (notation: $\Vdash \phi$) if it is valid on the class of all frames. The set of all formulas that are valid in a class of frames F is called the *logic* of F (notation: Λ_{F}). ⊣

Our definition of the logic of a frame class F (as the set of 'all' formulas that are valid on F) is underspecified: we did not say which collection of proposition letters Φ should be used to build formulas. But usually the precise form of this collection is irrelevant for our purposes. On the few occasions in this book where more precision is required, we will explicitly deal with the issue. (If the reader is worried about this, he or she may just fix a countable set Φ of proposition letters and define Λ_{F} to be $\{\phi \in Form(\tau, \Phi) \mid \mathsf{F} \Vdash \phi\}$.)

As will become abundantly clear in the course of the book, validity differs from truth in many ways. Here's a simple example. When a formula $\phi \vee \psi$ is true at a point w, this means that that either ϕ or ψ is true at w (the satisfaction definition tells us so). On the other hand, if $\phi \vee \psi$ is valid on a frame $\mathfrak{F}$, this does *not* mean that either ϕ or ψ is valid on $\mathfrak{F}$ ($p \vee \neg p$ is a simple counterexample).

Example 1.29 (i) The formula $\Diamond(p \vee q) \to (\Diamond p \vee \Diamond q)$ is valid on all frames. To see this, take any frame $\mathfrak{F}$ and state w in $\mathfrak{F}$, and let V be a valuation on $\mathfrak{F}$. We have to show that if $(\mathfrak{F}, V), w \Vdash \Diamond(p \vee q)$, then $(\mathfrak{F}, V), w \Vdash \Diamond p \vee \Diamond q$. So assume that $(\mathfrak{F}, V), w \Vdash \Diamond(p \vee q)$. Then, by definition there is a state v such that Rwv and $(\mathfrak{F}, V), v \Vdash p \vee q$. But, if $v \Vdash p \vee q$ then either $v \Vdash p$ or $v \Vdash q$. Hence either $w \Vdash \Diamond p$ or $w \Vdash \Diamond q$. Either way, $w \Vdash \Diamond p \vee \Diamond q$.

(ii) The formula $\Diamond\Diamond p \to \Diamond p$ is not valid on all frames. To see this we need to find a frame $\mathfrak{F}$, a state w in $\mathfrak{F}$, and a valuation on $\mathfrak{F}$ that falsifies the formula at w. So let $\mathfrak{F}$ be a three-point frame with universe $\{0, 1, 2\}$ and relation $\{(0, 1), (1, 2)\}$. Let V be any valuation on $\mathfrak{F}$ such that $V(p) = \{2\}$. Then $(\mathfrak{F}, V), 0 \Vdash \Diamond\Diamond p$, but $(\mathfrak{F}, V), 0 \nVdash \Diamond p$ since 0 is not related to 2.

(iii) But there is a class of frames on which $\Diamond\Diamond p \to \Diamond p$ is valid: the class of *transitive* frames. To see this, take any transitive frame $\mathfrak{F}$ and state w in $\mathfrak{F}$, and let V be a valuation on $\mathfrak{F}$. We have to show that if $(\mathfrak{F}, V), w \Vdash \Diamond\Diamond p$, then $(\mathfrak{F}, V), w \Vdash \Diamond p$. So assume that $(\mathfrak{F}, V), w \Vdash \Diamond\Diamond p$. Then by definition there are states u and v such that Rwu and Ruv and $(\mathfrak{F}, V), v \Vdash p$. But as R is transitive, it follows that Rwv, hence $(\mathfrak{F}, V), w \Vdash \Diamond p$.

(iv) As the previous example suggests, when additional constraints are imposed on frames, more formulas may become valid. For example, consider the frame depicted in Figure 1.2. On this frame the formula $\langle a \rangle p \to \langle b \rangle p$ is not valid; a countermodel is obtained by putting $V(p) = \{w_2\}$. Now, consider a frame satisfying the condition $R_a \subseteq R_b$; an example is depicted in Figure 1.7.

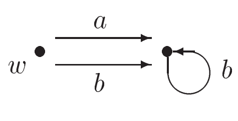


Fig. 1.7. A frame satisfying $R_a \subseteq R_b$.

On this frame it is impossible to refute the formula $\langle a \rangle p \to \langle b \rangle p$ at w, because a refutation would require the existence of a point u with $R_a wu$ and p true at u, but not $R_b wu$; but such points are forbidden when we insist that $R_a \subseteq R_b$.

This is a completely general point: in *every* frame $\mathfrak{F}$ of the appropriate similarity type, if $\mathfrak{F}$ satisfies the condition $R_a \subseteq R_b$, then $\langle a \rangle p \to \langle b \rangle p$ is valid in $\mathfrak{F}$. Moreover, the converse to this statement also holds: whenever $\langle a\rangle p \to \langle b\rangle p$ is valid on a given frame $\mathfrak{F}$, then the frame must satisfy the condition $R_a \subseteq R_b$. To use the terminology we will introduce in Chapter 3, the formula $\langle a\rangle p \to \langle b\rangle p$ *defines* the property that $R_a \subseteq R_b$.

(v) When interpreting the basic temporal language (see Example 1.25) we observed that arbitrary frames of the form (W, R_P, R_F) were uninteresting given the intended interpretation of F and P, and we insisted on interpreting them using a relation R and its converse. Interestingly, there is a sense in which the basic temporal language itself is strong enough to enforce the condition that the relation R_P is the converse of the relation R_F: such frames are *precisely* the ones which validate both the formulas $p \to GPp$ and $p \to HFp$; see Exercise 3.1.1.

(vi) The formula $Fq \to FFq$ is not valid on all frames. To see this we need to find a frame $\mathfrak{T} = (T, R)$, a state t in $\mathfrak{T}$, and a valuation on $\mathfrak{T}$ that falsifies this formula at t. So let $T = \{0, 1\}$, and let R be the relation $\{(0, 1)\}$. Let V be a valuation such that $V(p) = \{1\}$. Then $(\mathfrak{T}, V), 0 \Vdash Fp$, but obviously $(\mathfrak{T}, V), 0 \nVdash FFp$.

(vii) But there is a frame on which $Fp \to FFp$ is valid. As the universe of the frame take the set of all rational numbers $\mathbb{Q}$, and let the frame relation be the usual $<$-ordering on $\mathbb{Q}$. To show that $Fp \to FFp$ is valid on this frame, take any point t in it, and any valuation V such that $(\mathbb{Q}, <, V), t \Vdash Fp$; we have to show that $t \Vdash FFp$. But this is easy: as $t \Vdash Fp$, there exists a t' such that $t < t'$ and $t' \Vdash p$. Because we are working on the rationals, there must be an s with $t < s$ and $s < t'$ (for example, $(t + t')/2$). As $s \Vdash Fp$, it follows that $t \Vdash FFp$.

(viii) The special conditions demanded of PDL models also give rise to validities. For example, $\langle \pi_1 ; \pi_2\rangle p \leftrightarrow \langle \pi_1\rangle\langle \pi_2\rangle p$ is valid on any frame such that $R_{\pi_1;\pi_2} = R_{\pi_1} \circ R_{\pi_2}$, and in fact the converse is also true. The reader is asked to prove this in Exercise 3.1.2.

(ix) In our last example we consider arrow logic. We claim that in any square arrow frame $\mathfrak{S}_U$, the formula $\otimes(p \circ q) \to \otimes q \circ \otimes p$ is valid. For, let V be a valuation on $\mathfrak{S}_U$, and suppose that for some pair of points u, v in U, we have $(\mathfrak{S}_U, V), (u, v) \Vdash \otimes(p \circ q)$. It follows that $(\mathfrak{S}_U, V), (v, u) \Vdash p \circ q$, and hence, there must be a $w \in U$ for which $(\mathfrak{S}_U, V), (v, w) \Vdash p$ and $(\mathfrak{S}_U, V), (w, u) \Vdash q$. But then we have $(\mathfrak{S}_U, V), (w, v) \Vdash \otimes p$ and $(\mathfrak{S}_U, V), (u, w) \Vdash \otimes q$. This in turn implies that $(\mathfrak{S}_U, V), (u, v) \Vdash \otimes q \circ \otimes p$. $\dashv$

Exercises for Section 1.3

1.3.1 Show that when evaluating a formula ϕ in a model, the only relevant information in the valuation is the assignments it makes to the propositional letters actually occurring in ϕ. More precisely, let $\mathfrak{F}$ be a frame, and V and V' be two valuations on $\mathfrak{F}$ such that $V(p) = V'(p)$ for all proposition letters p in ϕ. Show that $(\mathfrak{F}, V) \Vdash \phi$ iff $(\mathfrak{F}, V') \Vdash \phi$. Work in the basic modal language. Do this exercise by *induction on the number of connectives* in ϕ (or as we usually put it, by *induction on* ϕ). (If you are unsure how to do this, glance ahead to Proposition 2.3 where such a proof is given in detail.)

1.3.2 Let $\mathfrak{N} = (\mathbb{N}, S_1, S_2)$ and $\mathfrak{B} = (\mathbb{B}, R_1, R_2)$ be the following frames for a modal similarity type with two diamonds $\Diamond_1$ and $\Diamond_2$. Here $\mathbb{N}$ is the set of natural numbers, $\mathbb{B}$ is the set of strings of 0s and 1s, and the relations are defined by

$$\begin{array}{rll} mS_1n & \text{iff} & n = m+1, \\ mS_2n & \text{iff} & m > n, \\ sR_1t & \text{iff} & t = s0 \text{ or } t = s1, \\ sR_2t & \text{iff} & t \text{ is a proper initial segment of } s. \end{array}$$

Which of the following formulas are valid on $\mathfrak{N}$ and $\mathfrak{B}$, respectively?

(a) $(\Diamond_1 p \wedge \Diamond_1 q) \to \Diamond_1(p \wedge q)$,
(b) $(\Diamond_2 p \wedge \Diamond_2 q) \to \Diamond_2(p \wedge q)$,
(c) $(\Diamond_1 p \wedge \Diamond_1 q \wedge \Diamond_1 r) \to (\Diamond_1(p \wedge q) \vee \Diamond_1(p \wedge r) \vee \Diamond_1(q \wedge r))$,
(d) $p \to \Diamond_1 \Box_2 p$,
(e) $p \to \Diamond_2 \Box_1 p$,
(f) $p \to \Box_1 \Diamond_2 p$,
(g) $p \to \Box_2 \Diamond_1 p$.

1.3.3 Consider the basic temporal language and the frames $(\mathbb{Z}, <)$, $(\mathbb{Q}, <)$ and $(\mathbb{R}, <)$ (the integer, rational, and real numbers, respectively, all ordered by the usual less-than relation $<$). In this exercise we use $\mathrm{E}\phi$ to abbreviate $P\phi \vee \phi \vee F\phi$, and $\mathrm{A}\phi$ to abbreviate $H\phi \wedge \phi \wedge G\phi$. Which of the following formulas are valid on these frames?

(a) $GGp \to p$,
(b) $(p \wedge Hp) \to FHp$,
(c) $(\mathrm{E}p \wedge \mathrm{E}\neg p \wedge \mathrm{A}(p \to Hp) \wedge \mathrm{A}(\neg p \to G\neg p)) \to \mathrm{E}(Hp \wedge G\neg p)$.

1.3.4 Show that every formula that has the form of a propositional tautology is valid. Further, show that $\Box(p \to q) \to (\Box p \to \Box q)$ is valid.

1.3.5 Show that each of the following formulas is *not* valid by constructing a frame $\mathfrak{F} = (W, R)$ that refutes it.

(a) $\Box\bot$,
(b) $\Diamond p \to \Box p$,
(c) $p \to \Box\Diamond p$,
(d) $\Diamond\Box p \to \Box\Diamond p$.

Find, for each of these formulas, a non-empty class of frames on which it is valid.

1.3.6 Show that the arrow formulas $\phi \circ (\psi \circ \chi) \leftrightarrow (\phi \circ \psi) \circ \chi$ and $1' \circ \phi \leftrightarrow \phi$ are valid in any square.

1.4 General Frames

At the level of models the fundamental concept is satisfaction. This is a relatively simple concept involving only a frame and a *single* valuation. By ascending to the level of frames we get a deeper grip on relational structures — but there is a price to pay. Validity lacks the concrete character of satisfaction, for it is defined in terms of *all* valuations on a frame. However there is an intermediate level: a *general frame* $(\mathfrak{F}, A)$ is a frame $\mathfrak{F}$ together with a restricted, but suitably well-behaved collection A of *admissible valuations*.

General frames are useful for at least two reasons. First, there may be application driven motivations to exclude certain valuations. For instance, if we were using $(\mathbb{N}, <)$ to model the temporal distribution of outputs from a computational device, it would be unreasonable to let valuations assign non recursively enumerable sets to propositional variables. But perhaps the most important reason to work with general frames is that they support a notion of validity that is mathematically simpler than the frame-based one, without losing too many of the concrete properties that make models so easy to work with. This 'simpler behavior' will only really become apparent when we discuss the algebraic perspective on completeness theory in Chapter 5. It will turn out that there is a fundamental and universal *completeness result* for general frame validity, something that the frame semantics lacks. Moreover, we will discover that general frames are essentially a set-theoretic representation of *boolean algebras with operators*. Thus, the A in (W, R, A) stands not only for *Admissible*, but also for *Algebra*.

So what is a 'suitably well-behaved collection of valuations'? It simply means a collection of valuations closed under the set-theoretic operations corresponding to our connectives and modal operators. Now, fairly obviously, the boolean connectives correspond to the boolean operations of union, relative complement, and so on — but what operations on sets do modalities correspond to? Here is the answer.

Let us first consider the basic modal similarity type with one diamond. Given a frame $\mathfrak{F} = (W, R)$, let $m_\diamond$ be the following operation on the power set of W:

$$m_\diamond(X) = \{w \in W \mid Rwx \text{ for some } x \in X\ \}.$$

Think of $m_\diamond(X)$ as the set of states that 'see' a state in X. This operation corresponds to the diamond in the sense that for any valuation V and any formula ϕ:

$$V(\diamond\phi) = m_\diamond(V(\phi)).$$

Moving to the general case, we obtain the following definition.

Definition 1.30 Let τ be a modal similarity type, and $\mathfrak{F} = (W, R_\triangle)_{\triangle\in\tau}$ a τ-frame. For $\triangle \in \tau$ we define the following function $m_\triangle$ on the power set of W:

$$\begin{aligned} m_\triangle(X_1, \ldots, X_n) \;\; = \;\; & \{w \in W \mid \text{ there are } w_1, \ldots, w_n \in W \text{ such that} \\ & R_\triangle w w_1 \ldots w_n \text{ and } w_i \in X_i, \text{ for all } i = 1, \ldots, n.\} \end{aligned} \quad \dashv$$

Example 1.31 Let $\otimes$ be the converse operator of arrow logic, and consider a square frame $\mathfrak{S}_U$. Note that $m_\otimes$ is the following operation:

$$m_\otimes(X) \quad = \quad \{a \in U^2 \mid Rax \text{ for some } x \in X\ \}.$$

But by the rather special nature of R this boils down to

$$\begin{aligned} m_\otimes(X) &= \{(a_0, a_1) \in U^2 \mid a_0 = x_1 \text{ and } a_1 = x_0 \text{ for some } (x_0, x_1) \in X\ \}, \\ &= \{(x_1, x_0) \in U^2 \mid (x_0, x_1) \in X\}. \end{aligned}$$

In other words, $m_\otimes(X)$ is nothing but the *converse* of the binary relation X. $\dashv$

Definition 1.32 (General Frames) Let τ be a modal similarity type. A *general τ-frame* is a pair $(\mathfrak{F}, A)$ where $\mathfrak{F} = (W, R_\triangle)_{\triangle\in\tau}$ is a τ-frame, and A is a non-empty collection of subsets of W closed under the following operations:

(i) union: if $X, Y \in A$ then $X \cup Y \in A$.
(ii) relative complement: if $X \in A$, then $W \setminus X \in A$.
(iii) modal operations: if $X_1, \ldots, X_n \in A$, then $m_\triangle(X_1, \ldots, X_n) \in A$ for all $\triangle \in \tau$.

A *model based on a general frame* is a triple $(\mathfrak{F}, A, V)$ where $(\mathfrak{F}, A)$ is a general frame and V is a valuation satisfying the constraint that for each proposition letter p, $V(p)$ is an element of A. Valuations satisfying this constraint are called *admissible* for $(\mathfrak{F}, A)$. $\dashv$

It follows immediately from the first two clauses of the definition that both the empty set and the universe of a general frame are always admissible. Note that an ordinary frame $\mathfrak{F} = (W, R_\triangle)_{\triangle\in\tau}$ can be regarded as a general frame where $A = \mathcal{P}(W)$ (that is, a general frame in which all valuations are admissible). Also, note that if a valuation V is admissible for a general frame $(\mathfrak{F}, A)$, then the closure conditions listed in Definition 1.32 guarantee that $V(\phi) \in A$, for all formulas ϕ. In short, a set of admissible valuations A is a 'logically closed' collection of information assignments.

Definition 1.33 A formula ϕ is *valid at a state w in a general frame* $(\mathfrak{F}, A)$ (notation: $(\mathfrak{F}, A), w \Vdash \phi$) if ϕ is true at w in every admissible model $(\mathfrak{F}, A, V)$ on $(\mathfrak{F}, A)$; and ϕ is *valid in a general frame* $(\mathfrak{F}, A)$ (notation: $(\mathfrak{F}, A) \Vdash \phi$) if ϕ is true at every state in every admissible model $(\mathfrak{F}, A, V)$ on $(\mathfrak{F}, A)$.

A formula ϕ is *valid on a class of general frames* G (notation: $\mathsf{G} \Vdash \phi$) if it is valid on every general frame $(\mathfrak{F}, A)$ in G. Finally, if ϕ is valid on the class of all general frames we say that it is *g-valid* and write $\Vdash_g \phi$. We will learn in Chapter 4 (see Exercise 4.1.1) that a formula ϕ is valid if and only if it is g-valid. $\dashv$

Clearly, for any frame $\mathfrak{F}$, if $\mathfrak{F} \Vdash \phi$ then for any collection of admissible assignments A on $\mathfrak{F}$, we have $(\mathfrak{F}, A) \Vdash \phi$ too. The converse does not hold. Here is a counterexample that will be useful in Chapter 4.

Example 1.34 Consider the McKinsey formula, $\Box\Diamond p \to \Diamond\Box p$. It is easy to see that the McKinsey formula is *not* valid on the frame $(\mathbb{N}, <)$, for we obtain a countermodel by choosing a valuation for p that lets the truth value of p alternate infinitely often (for instance, by letting $V(p)$ be the collection of even numbers).

However there is a general frame based on $(\mathbb{N}, <)$ in which the McKinsey formula *is* valid. First some terminology: a set is *co-finite* if its complement is finite. Now consider the general frame $\mathfrak{f} = (\mathbb{N}, <, A)$, where A is the collection of all finite and co-finite sets. We leave it as an exercise to show that $\mathfrak{f}$ satisfies all the constraints of Definition 1.32; see Exercise 1.4.5.

To see that the McKinsey formula is indeed valid on $\mathfrak{f}$, let V be an admissible valuation, and let $n \in \mathbb{N}$. If $(\mathfrak{f}, V), n \Vdash \Box\Diamond p$, then $V(p)$ must be co-finite (why?), hence for some k every state $l \geq k$ is in $V(p)$. But this means that $(\mathfrak{f}, V), n \Vdash \Diamond\Box p$, as required. $\dashv$

Although we will make an important comment about general frames in Section 3.2, and use them to help prove an incompleteness result in Section 4.4, we will not really be in a position to grasp their significance until Chapter 5, when we introduce boolean algebras with operators. Until then, we will concentrate on modal languages as tools for talking about models and frames.

Exercises for Section 1.4

1.4.1 Define, analogous to $m_\Diamond$, an operation $m_\Box$ on the power set of a frame such that for an arbitrary modal formula ϕ and an arbitrary valuation V we have that $m_\Box(V(\phi)) = V(\Box\phi)$. Extend this definition to the dual of a polyadic modal operator.

1.4.2 Consider the basic modal formula $\Diamond p \to \Box p$.

(a) Construct a frame $\mathfrak{F} = (W, R)$ and a general frame $\mathfrak{f} = (\mathfrak{F}, A)$ such that $\mathfrak{F} \nVdash \Diamond p \to \Box p$, but $\mathfrak{f} \Vdash \Diamond p \to \Box p$.
(b) Construct a general frame $(\mathfrak{F}, A)$ and a valuation V on $\mathfrak{F}$ such that $(\mathfrak{F}, A) \nVdash \Diamond p \to \Box p$, but $(\mathfrak{F}, V) \Vdash \Diamond p \to \Box p$.

1.4.3 Show that if B is any collection of valuations over some frame $\mathfrak{F}$, then there is a smallest general frame $(\mathfrak{F}, A)$ such that $B \subseteq A$. ('Smallest' means that for any general frame $(\mathfrak{F}, A')$ such that $B \subseteq A'$, $A \subseteq A'$.)

1.4.4 Show that for square arrow frames, the operation $m_\circ$ is nothing but *composition* of two binary relations. What is $m_{1'}$?

1.4.5 Consider the basic modal language, and the general frame $\mathfrak{f} = (\mathbb{N}, <, A)$, where A is the collection of all finite and co-finite sets. Show that $\mathfrak{f}$ is a general frame.

1.4.6 Consider the structure $\mathfrak{g} = (\mathbb{N}, C, A)$ where A is the collection of finite and cofinite subsets of $\mathbb{N}$, and C is defined by

$$Cn_1n_2n_3 \text{ iff } n_1 \leq n_2 + n_3 \text{ and } n_2 \leq n_3 + n_1 \text{ and } n_3 \leq n_1 + n_2.$$

If C is the accessibility relation of a dyadic modal operator, show that $\mathfrak{g}$ is a general frame.

1.4.7 Let $\mathfrak{M} = (\mathfrak{F}, V)$ be some modal model. Prove that the structure

$$(\mathfrak{F}, \{V(\phi) \mid \phi \text{ is a formula }\})$$

is a general frame.

1.5 Modal Consequence Relations

While the idea of validity in frames (and indeed, validity in general frames) gives rise to logically interesting formulas, so far we have said nothing about what *logical consequence* might mean for modal languages. That is, we have not explained what it means for a set of modal formulas Σ to logically entail a modal formula ϕ.

This we will now do. In fact, we will introduce *two* families of consequence relations: a local one and a global one. Both families will be defined *semantically*; that is, in terms of classes of structures. We will define these relations for all three kinds of structures we have introduced, though in practice we will be primarily interested in semantic consequence over frames. Before going further, a piece of terminology. If S is a class of models, then *a model from* S is simply a model $\mathfrak{M}$ in S. On the other hand, if S is a class of frames (or a class of general frames) then a model from S is a model based on a frame (general frame) in S.

What is a modally reasonable notion of logical consequence? Two things are fairly clear. First, it seems sensible to hold on to the familiar idea that a relation of semantic consequence holds when the truth of the premises guarantees the truth of the conclusion. Second, it should be clear that the inferences we are entitled to draw will depend on the class of structures we are working with. (For example, different inferences will be legitimate on transitive and intransitive frames.) Thus our definition of consequence will have to be parametric: it must make reference to a class of structures S.

Here's the standard way of meeting these requirements. Suppose we are working with a class of structures S. Then, for a formula ϕ (the *conclusion*) to be a logical consequence of Σ (the *premises*) we should insist that whenever Σ is true at some point in some model from S, then ϕ should also be true in that same model *at the same point*. In short, this definition demands that the maintenance of truth should be guaranteed *point to point* or *locally*.

Definition 1.35 (Local Semantic Consequence) Let τ be a similarity type, and let S be a class of structures of type τ (that is a class of models, a class of frames, or a class of general frames of this type). Let Σ and ϕ be a set of formulas and a single formula from a language of type τ. We say that ϕ is a *local semantic consequence of Σ over* S (notation: $\Sigma \Vdash_{\mathsf{S}} \phi$) if for all models $\mathfrak{M}$ from S, and all points w in $\mathfrak{M}$, if $\mathfrak{M}, w \Vdash \Sigma$ then $\mathfrak{M}, w \Vdash \phi$. $\dashv$

Example 1.36 Suppose that we are working with Tran, the class of transitive frames. Then:

$$\{\Diamond\Diamond p\} \Vdash_{\mathsf{Tran}} \Diamond p.$$

On the other hand, $\Diamond p$ is *not* a local semantic consequence of $\{\Diamond\Diamond p\}$ over the class of *all* frames. $\dashv$

Local consequence is the notion of logical entailment explored in this book, but it is by no means the only possibility. Here's an obvious variant.

Definition 1.37 (Global Semantic Consequence) Let τ, S, Σ and ϕ be as in Definition 1.35. We say that ϕ is a *global semantic consequence of Σ over* S (notation: $\Sigma \Vdash^g_{\mathsf{S}} \phi$) if and only if for all structures $\mathfrak{S}$ in S, if $\mathfrak{S} \Vdash \Sigma$ then $\mathfrak{S} \Vdash \phi$. (Here, depending on the kind of structures S contains, $\Vdash$ denotes either validity in a frame, validity in a general frame, or global truth in a model.) $\dashv$

Again, this definition hinges on the idea that premises guarantee conclusions, but here the guarantee covers *global* notions of correctness.

Example 1.38 The local and global consequence relations are different. Consider the formulas p and $\Box p$. It is easy to see that p does not locally imply $\Box p$ — indeed, that this entailment should *not* hold is pretty much the essence of locality. On the other hand, suppose that we consider a model $\mathfrak{M}$ where p is globally true. Then p certainly holds at all successors of all states, so $\mathfrak{M} \Vdash \Box p$, and so $p \Vdash^g \Box p$. $\dashv$

Nonetheless, there is a systematic connection between the two consequence relations, as the reader is asked to show in Exercise 1.5.3.

Exercises for Section 1.5

1.5.1 Let K be a class of frames for the basic modal similarity type, and let $\mathsf{M}(\mathsf{K})$ denote the class of models based on a frame in K. Prove that $p \Vdash^g_{\mathsf{M}(\mathsf{K})} \Diamond p$ iff $\mathsf{K} \models \forall x \exists y\, Ryx$ (every point has a predecessor).

Does this equivalence hold as well if we work with $\Vdash^g_{\mathsf{K}}$ instead?

1.5.2 Let M denote the class of all models, and F the class of all frames. Show that if $\Sigma \Vdash^g_{\mathsf{M}} \phi$ then $\Sigma \Vdash^g_{\mathsf{F}} \phi$, but that the converse is false.

1.5.3 Let Σ be a set of formulas in the basic modal language, and let F denote the class of all frames. Show that $\Sigma \Vdash^g_{\mathsf{F}} \phi$ iff $\{\Box^n \sigma \mid \sigma \in \Sigma, n \in \omega\} \Vdash_{\mathsf{F}} \phi$.

1.5.4 Again, let F denote the class of all frames. Show that the local consequence relation does have the deduction theorem: $\phi \Vdash \psi$ iff $\Vdash \phi \to \psi$, but the global one does not. However, show that on the class Tran of transitive frames we have that $\phi \Vdash^g_{\mathsf{Tran}} \psi$ iff $\Vdash^g_{\mathsf{Tran}} \Box\phi \to \psi$.

1.6 Normal Modal Logics

Till now our discussion has been largely *semantic*; but logic has an important *syntactic* dimension, and our discussion raises some obvious questions. Suppose we are interested in a certain class of frames F: are there syntactic mechanisms capable of generating Λ_{F}, the formulas valid on F? And are such mechanisms capable of coping with the associated semantic consequence relation? The modal logician's response to such questions is embodied in the concept of a *normal modal logic*.

A normal modal logic is simply a set of formulas satisfying certain syntactic closure conditions. Which conditions? We will work towards the answer by defining a Hilbert-style axiom system called **K**. **K** is the 'minimal' (or 'weakest') system for reasoning about frames; stronger systems are obtained by adding extra axioms. We discuss **K** in some detail, and then, at the end of the section, define normal modal logics. By then, the reader will be in a position to see that the definition is a more-or-less immediate abstraction from what is involved in Hilbert-style approaches to modal proof theory. We will work in the basic modal language.

Definition 1.39 A **K**-*proof* is a finite sequence of formulas, each of which is an *axiom*, or follows from one or more earlier items in the sequence by applying a *rule of proof*. The axioms of **K** are *all instances of propositional tautologies* plus:

(K) $\Box(p \to q) \to (\Box p \to \Box q)$
(Dual) $\Diamond p \leftrightarrow \neg\Box\neg p$.

The rules of proof of **K** are:

- *Modus ponens*: given ϕ and $\phi \to \psi$, prove ψ.
- *Uniform substitution*: given ϕ, prove θ, where θ is obtained from ϕ by uniformly replacing proposition letters in ϕ by arbitrary formulas.
- *Generalization*: given ϕ, prove $\Box\phi$.

A formula ϕ is **K**-*provable* if it occurs as the last item of some **K**-proof, and if this is the case we write $\vdash_{\mathbf{K}} \phi$. $\dashv$

Some comments. Tautologies may contain modalities (for example, $\Diamond q \vee \neg\Diamond q$ is a tautology, as it has the same form as $p \vee \neg p$). As tautologies are valid on all frames (Exercise 1.3.4), they are a safe starting point for modal reasoning. Our decision to add *all* propositional tautologies as axioms is an example of axiomatic overkill;

we could have chosen a small set of tautologies capable of generating the rest via the rules of proof, but this refinement is of little interest for our purposes.

Modus ponens is probably familiar to all our readers, but there are two important points we should make. First, *modus ponens preserves validity*. That is, if $\Vdash \phi$ and $\Vdash \phi \to \psi$ then $\Vdash \psi$. Given that we want to reason about frames, this property is crucial. Note, however, that modus ponens also preserves two further properties, namely *global truth* (if $\mathfrak{M} \Vdash \phi$ and $\mathfrak{M} \Vdash \phi \to \psi$ then $\mathfrak{M} \Vdash \psi$) and *satisfiability* (if $\mathfrak{M}, w \Vdash \phi$ and $\mathfrak{M}, w \Vdash \phi \to \psi$ then $\mathfrak{M}, w \Vdash \psi$). That is, modus ponens is not only a correct rule for reasoning about frames, it is also a correct rule for reasoning about models, both globally and locally.

Uniform substitution should also be familiar. It mirrors the fact that validity abstracts away from the effects of particular assignments: if a formula is valid, this cannot be because of the particular value its propositional symbols have, thus we should be free to uniformly replace these symbols with any other formula whatsoever. And indeed, as the reader should check, *uniform substitution preserves validity*. Note, however, that it does *not* preserve either global truth or satisfiability. (For example, q is obtainable from p by uniform substitution, but just because p is globally true in some model, it does *not* follow that q is too!) In short, uniform substitution is strictly a tool for generating new validities from old.

That's the classical core of our Hilbert system, so let's turn to the the genuinely *modal* axioms and rules of proof. First the axioms. The K axiom is the fundamental one. It is clearly *valid* (as the reader who has not done Exercise 1.3.4 should now check) but why is it a useful addition to our Hilbert system?

K is sometimes called the *distribution axiom*, and is important because it lets us transform $\Box(\phi \to \psi)$ (a boxed formula) into $\Box\phi \to \Box\psi$ (an implication). This box-over-arrow distribution enables further purely propositional reasoning to take place. For example, suppose we are trying to prove $\Box\psi$, and have constructed a proof sequence containing both $\Box(\phi \to \psi)$ and $\Box\phi$. If we could apply modus ponens under the scope of the box, we would have proved $\Box\psi$. And this is what distribution lets us do: as **K** contains the axiom $\Box(p \to q) \to (\Box p \to \Box q)$, by uniform substitution we can prove $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$. But then a first application of modus ponens proves $\Box\phi \to \Box\psi$, and a second proves $\Box\psi$ as desired.

The Dual axiom obviously reflects the duality of $\Diamond$ and $\Box$; nonetheless, readers familiar with other discussions of **K** (many of which have K as the sole modal axiom) may be surprised at its inclusion. Do we really need it? Yes, we do. In this book, $\Diamond$ is primitive and $\Box$ is an abbreviation. Thus our K axiom is really shorthand for $\neg\Diamond\neg(p \to q) \to (\neg\Diamond\neg p \to \neg\Diamond\neg q)$. We need a way to maneuver around these negations, and this is the *syntactic* role that Dual plays. (Incidentally had we chosen $\Box$ as our primitive operator, Dual would *not* have been required.) We prefer working with a primitive $\Diamond$ (apart from anything else, it is more convenient for the algebraic work of Chapter 5) and do not mind adding Dual as an extra axiom. Dual, of course, is valid.

It only remains to discuss the modal rule of proof: *generalization* (another common name for it is *necessitation*). Generalization 'modalizes' provable formulas by stacking boxes in front. Roughly speaking, while the K axiom lets us apply classical reasoning inside modal contexts, necessitation creates new modal contexts for us to work with; modal proofs arise from the interplay of these two mechanisms.

Note that generalization preserves validity: if it is impossible to falsify ϕ, then obviously we will never be able to falsify ϕ at any accessible state! Similarly, generalization preserves *global* truth. But it *does not* preserve satisfaction: just because p is true in some state, we cannot conclude that p is true at all accessible states.

K is the minimal modal Hilbert system in the following sense. As we have seen, its axioms are all valid, and all three rules of inference preserve validity, hence all **K**-provable formulas are valid. (To use the terminology introduced in Definition 4.9, **K** is *sound* with respect to the class of all frames.) Moreover, as we will prove in Theorem 4.23, the converse is also true: *if a basic modal formula is valid, then it is* **K***-provable*. (That is, **K** is *complete* with respect to the class of all frames.) In short, **K** generates precisely the valid formulas.

Example 1.40 The formula $(\Box p \wedge \Box q) \rightarrow \Box(p \wedge q)$ is valid on any frame, so it should be **K**-provable. And indeed, it is. To see this, consider the following sequence of formulas:

1.	$\vdash p \rightarrow (q \rightarrow (p \wedge q))$	Tautology
2.	$\vdash \Box(p \rightarrow (q \rightarrow (p \wedge q)))$	Generalization: 1
3.	$\vdash \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$	K axiom
4.	$\vdash \Box(p \rightarrow (q \rightarrow (p \wedge q)) \rightarrow (\Box p \rightarrow \Box(q \rightarrow (p \wedge q)))$	Uniform Substitution: 3
5.	$\vdash \Box p \rightarrow \Box(q \rightarrow (p \wedge q))$	Modus Ponens: 2, 4
6.	$\vdash \Box(q \rightarrow (p \wedge q)) \rightarrow (\Box q \rightarrow \Box(p \wedge q))$	Uniform Substitution: 3
7.	$\vdash \Box p \rightarrow (\Box q \rightarrow \Box(p \wedge q))$	Propositional Logic: 5, 6
8.	$\vdash (\Box p \wedge \Box q) \rightarrow \Box(p \wedge q)$	Propositional Logic: 7

Strictly speaking, this sequence is *not* a **K**-proof — it is a subsequence of the proof consisting of the most important items. The annotations in the right-hand column should be self-explanatory; for example 'Modus Ponens: 2, 4' labels the formula obtained from the second and fourth formulas in the sequence by applying modus ponens. To obtain the full proof, fill in the items that lead from line 6 to 8. $\dashv$

Remark 1.41 Warning: there is a pitfall that is *very* easy to fall into if you are used to working with natural deduction systems: we *cannot* freely make and discharge assumptions in the Hilbert system **K**. The following 'proof' shows what can go wrong if we do:

1.	p	Assumption
2.	$\Box p$	Generalization: 1
3.	$p \to \Box p$	Discharge assumption

So we have 'proved' $p \to \Box p$! This is obviously wrong: this formula is *not* valid, hence it is *not* **K**-provable. And it should be clear where we have gone wrong: we *cannot* use assumptions as input to generalization, for, as we have already remarked, this rule does *not* preserve satisfiability. Generalization is there to enable us to generate new validities from old. It is not a local rule of inference. $\dashv$

For many purposes, **K** is too weak. If we are interested in transitive frames, we would like a proof system which reflects this. For example, we know that $\Diamond\Diamond p \to \Diamond p$ is valid on all transitive frames, so we would want a proof system that generates this formula; **K** does not do this, for $\Diamond\Diamond p \to \Diamond p$ is not valid on all frames.

But we can extend **K** to cope with many such restrictions by adding extra axioms. For example, if we enrich **K** by adding $\Diamond\Diamond p \to \Diamond p$ as an axiom, we obtain the Hilbert-system called **K4**. As we will show in Theorem 4.27, **K4** is sound and complete with respect to the class of all transitive frames (that is, it generates *precisely* the formulas valid on transitive frames). More generally, given any set of modal formulas Γ, we are free to add them as extra axioms to **K**, thus forming the axiom system $\mathbf{K}\boldsymbol{\Gamma}$. As we will learn in Chapter 4, in many important cases it is possible to characterize such extensions in terms of frame validity.

One final issue remains to be discussed: do such axiomatic extensions of **K** give us a grip on semantic consequence, and in particular, the local semantic consequence relation over classes of frames (see Definition 1.35)?

In many important cases they do. Here's the basic idea. Suppose we are interested in transitive frames, and are working with **K4**. We capture the notion of local consequence over transitive frames in **K4** as follows. Let Σ be a set of formulas, and ϕ a formula. Then we say that ϕ is a local *syntactic* consequence of Σ in **K4** (notation: $\Sigma \vdash_{\mathbf{K4}} \phi$) if and only if there is some finite subset $\{\sigma_1, \ldots, \sigma_n\}$ of Σ such that $\vdash_{\mathbf{K4}} \sigma_1 \wedge \cdots \wedge \sigma_n \to \phi$. In Theorem 4.27 we will show that

$$\Sigma \vdash_{\mathbf{K4}} \phi \text{ iff } \Sigma \Vdash_{\mathsf{Tran}} \phi,$$

where $\Vdash_{\mathsf{Tran}}$ denotes local semantic consequence over transitive frames. In short, we have reduced the local *semantic* consequence relation over transitive frames to provability in **K4**.

Definition 1.42 (Normal Modal Logics) A *normal modal logic* Λ is a set of formulas that contains all tautologies, $\Box(p \to q) \to (\Box p \to \Box q)$, and $\Diamond p \leftrightarrow \neg\Box\neg p$, and that is closed under *modus ponens*, *uniform substitution* and *generalization*. We call the smallest normal modal logic **K**. $\dashv$

This definition is a direct abstraction from the ideas underlying modal Hilbert systems. It throws away all talk of proof sequences and concentrates on what is really essential: the presence of axioms and closure under the rules of proof.

We will rarely mention Hilbert systems again: we prefer to work with the more abstract notion of normal modal logics. For a start, although the two approaches are equivalent (see Exercise 1.6.6), it is simpler to work with the set-theoretical notion of membership than with proof sequences. More importantly, in Chapters 4 and 5 we will prove results that link the semantic and syntactic perspectives on modal logic. These results will hold for *any* set of formulas fulfilling the normality requirements. Such a set might be the formulas generated by a Hilbert-style proof system — but it could just as well be the formulas provable in a natural-deduction system, a sequent system, a tableaux system, or a display calculus. Finally, the concept of a normal modal logic makes good semantic sense: for any class of frames F, we have that Λ_{F}, the set of formulas valid on F, is a normal modal logic; see Exercise 1.6.7.

Exercises for Section 1.6

1.6.1 Give **K**-proofs of $(\Box p \wedge \Diamond q) \to \Diamond(p \wedge q)$ and $\Diamond(p \vee q) \leftrightarrow (\Diamond p \vee \Diamond q)$.

1.6.2 Let ϕ^- be the 'demodalized' version of a modal formula ϕ; that is, ϕ^- is obtained from ϕ by simply erasing all diamonds. Prove that ϕ^- is a propositional tautology whenever ϕ is **K**-provable. Conclude that not every modal formula is **K**-provable.

1.6.3 The axiom system known as **S4** is obtained by adding the axiom $p \to \Diamond p$ to **K4**. Show that $\not\vdash_{\mathbf{S4}} p \to \Box\Diamond p$; that is, show that **S4** does *not* prove this formula. (Hint: find an appropriate class of frames for which **S4** is *sound*.) If we add this formula as an axiom to **S4** we obtain the system called **S5**. Give an **S5**-proof of $\Diamond\Box p \to \Box p$.

1.6.4 Try adapting **K** to obtain a minimal Hilbert system for the basic temporal language. Does your system cope with the fact that we only interpret this language on bidirectional frames? Then try and define a minimal Hilbert system for the language of propositional dynamic logic.

1.6.5 This exercise is only for readers who like syntactical manipulations and have a lot of time to spare. **KL** is the axiomatization obtained by adding the Löb formula $\Box(\Box p \to p) \to \Box p$ as an extra axiom to **K**. Try and find a **KL** proof of $\Box p \to \Box\Box p$. That is, show that **KL** = **KL4**.

1.6.6 In Chapter 4 we will use $\mathbf{K\Gamma}$ to denote the smallest normal modal logic containing Γ; the point of the present exercise is to relate this notation to our discussion of Hilbert systems. So (as discussed above) suppose we form the axiom system $\mathbf{K\Gamma}$ by adding as axioms all the formulas in Γ to **K**. Show that the *Hilbert system* $\mathbf{K\Gamma}$ proves precisely the formulas contained in the *normal modal logic* $\mathbf{K\Gamma}$.

1.6.7 Let F be a class of frames. Show that Λ_{F} is a normal modal logic.

1.7 Historical Overview

The ideas introduced in this chapter have a long history. They evolved as responses to particular problems and challenges, and knowing something of the context in which they arose will make it easier to appreciate why they are considered important, and the way they will be developed in subsequent chapters. Some of the discussion that follows may not be completely accessible at this stage. If so, don't worry. Just note the main points, and try again once you have explored the chapters that follow.

We find it useful to distinguish three phases in the development of modal logic: the *syntactic* era, the *classical* era, and the *modern* era. Roughly speaking, most of the ideas introduced in this chapter stem from the classical era, and the remainder of the book will explore them from the point of view of the modern era.

The syntactic era (1918–1959)

We have opted for 1918, the year that C.I. Lewis published his *Survey of Symbolic Logic* [306], as the birth of modal logic as a mathematical discipline. Lewis was certainly not the first to consider modal reasoning, indeed he was not even the first to construct symbolic systems for this purpose: Hugh MacColl, who explored the consequences of enriching propositional logic with operators ϵ ('it is certain that') and η ('it is impossible that') seems to have been the first to do that (see his book *Symbolic Logic and its Applications* [312], and for an overview of his work, see [373]). But MacColl's work is firmly rooted in the 19-th century algebraic tradition of logic (well-known names in this tradition include Boole, De Morgan, Jevons, Peirce, Schröder, and Venn), and linking MacColl's contributions to contemporary concerns is a non-trivial scholarly task. The link between Lewis's work and contemporary modal logic is more straightforward.

In his 1918 book, Lewis extended propositional calculus with a unary modality I ('it is impossible that') and defined the binary modality $\phi \prec \psi$ (ϕ *strictly implies* ψ) to be $\mathrm{I}(\phi \wedge \neg\psi)$. Strict implication was meant to capture the notion of logical entailment, and Lewis presented a $\prec$-based axiom system. Lewis and Langford's joint book *Symbolic Logic* [307], published in 1932, contains a more detailed development of Lewis' ideas. Here $\Diamond$ ('it is possible that') is primitive and $\phi \prec \psi$ is defined to be $\neg\Diamond(\phi \wedge \neg\psi)$. Five axiom systems of ascending strength, **S1**–**S5**, are discussed; **S3** is equivalent to Lewis' system of 1918, and only **S4** and **S5** are normal modal logics. Lewis' work sparked interest in the idea of 'modalizing' propositional logic, and there were many attempts to axiomatize such concepts as obligation, belief and knowledge. Von Wright's monograph *An Essay in Modal Logic* [456] is an important example of this type of work.

But in important respects, Lewis' work seems strange to modern eyes. For a start, his axiomatic systems are not modular. Instead of extending a base system of propositional logic with specifically modal axioms (as we did in this chapter when we defined **K**), Lewis defines his axioms directly in terms of $\prec$. The modular approach to modal Hilbert systems is due to Kurt Gödel. Gödel [181] showed that (propositional) intuitionistic logic could be translated into **S4** in a theorem-preserving way. However instead of using the Lewis and Langford axiomatization, Gödel took $\Box$ as primitive and formulated **S4** in the way that has become standard: he enriched a standard system for classical propositional logic with the rule of generalization, the K axiom, and the additional axioms ($\Box p \to p$ and $\Box p \to \Box\Box p$).

But the fundamental difference between current modal logic and the work of Lewis and his contemporaries is that the latter is essentially *syntactic*. Propositional logic is enriched with some new modality. By considering various axioms, the logician tries to pin down the logic of the intended interpretation. This simple view of logical modeling has its attractions, but is open to serious objections. First, there are technical difficulties. Suppose we have several rival axiomatizations of some concept. Forget for now the problem of judging which is the best, for there is a more basic difficulty: how can we tell if they are really different? If we only have access to syntactic ideas, proving that two Hilbert-systems generate different sets of formulas can be extremely difficult. Indeed, even showing syntactically that two Hilbert systems generate the *same* set of formulas can be highly non-trivial (recall Exercise 1.6.5).

Proving distinctness theorems was standard activity in the syntactic era; for instance, Parry [359] showed that **S2** and **S3** are distinct, and papers addressing such problems were common till the late 1950s. Algebraic methods were often used to prove distinctness. The propositional symbols would be viewed as denoting the elements of some algebra, and complex formulas interpreted using the algebraic operations. Indeed, algebras were the key tool driving the technical development of the period. For example, McKinsey [328] used them to analyze **S2** and **S4** and show their decidability; McKinsey and Tarski [330], McKinsey [329], and McKinsey and Tarski [331] extended this work in a variety of directions (giving, among other things, a topological interpretation of **S4**); while Dummett and Lemmon [125] built on this work to isolate and analyze **S4.2** and **S4.3**, two important normal logics between **S4** and **S5**. But for all their technical utility, algebraic methods seemed of limited help in providing reliable intuitions about modal languages and their associated logics. Sometimes algebraic elements were viewed as multiple truth values. But Dugundji [124] showed that no logic between **S1** and **S5** could be viewed as an n-valued logic for *finite* n, so the multi-valued perspective on modal logic was not suited as a reliable source of insight.

The lack of a natural semantics brings up a deeper problem facing the syntactic approach: how do we know we have considered all the relevant possibilities? Nowadays the normal logic **T** (that is, **K** enriched with the axiom $p \to \Diamond p$) would be considered a fundamental logic of possibility; but Lewis overlooked **T** (it is intermediate between **S2** and **S4** and neither contains nor is contained by **S3**). Moreover, although Lewis did isolate two logics still considered important (namely **S4** and **S5**), how could he claim that either system was, in any interesting sense, *complete*? Perhaps there are important axioms missing from both systems? The existence of so many competing logics should make us skeptical of claims that it is easy to find all the relevant axioms and rules; and without precise, intuitively acceptable, criteria of what the the reasonable logics are (in short, the type of criteria a decent semantics provides us with) we have no reasonable basis for claiming success.

For further discussion of the work of this period, the reader should consult the historical section of Bull and Segerberg [73]). We close our discussion of the syntactic era by noting three lines of work that anticipate later developments: Carnap's state-description semantics, Prior's work on temporal logic, and the Jónsson and Tarski Representation Theorem for boolean algebras with operators.

A *state description* is simply a collection of propositional letters. (Actually, Carnap used state descriptions in his pioneering work on first-order modal logic, so a state for Carnap could be a set of first-order formulas.) If S is a collection of state descriptions, and $s \in S$, then a propositional symbol p is satisfied at s if and only $p \in s$. Boolean operators are interpreted in the obvious way. Finally, $\Diamond\phi$ is satisfied at $s \in S$ if and only if there is some $s' \in S$ such that s' satisfies ϕ. (See, for example, Carnap [83, 84].)

Carnap's interpretation of $\Diamond\phi$ in state descriptions is strikingly close to the idea of satisfaction in models. However one crucial idea is missing: the use of an *explicit* relation R over state descriptions. In Carnap's semantics, satisfaction for $\Diamond$ is defined in terms of membership in S (in effect, R is taken to be $S \times S$). This implicit fixing of R reduces the utility of his semantics: it yields a semantics for one fixed interpretation of $\Diamond$, but deprives us of the vital parameter needed to map logical options.

Arthur Prior founded temporal logic (or as he called it, *tense logic*) in the early 1950s. He invented the basic temporal language and many other temporal languages, both modal and non-modal. Like most of his contemporaries, Prior viewed the axiomatic exploration of concepts as one of the logician's key tasks. But there the similarity ends: his writings are packed with an extraordinary number of semantic ideas and insights. By 1955 Prior had interpreted the basic modal language in models based on $(\omega, <)$ (see Prior [368], and Chapter 2 of Prior [369]), and used what would now be called soundness arguments to distinguish logics. Moreover, the relative expressivity of modal and classical languages (such as the Prior-Meredith U-calculus [333]) is a constant theme of his writings; indeed, much of his work anticipates later work in correspondence theory and extended modal logic. His work is hard to categorize, and impossible to summarize, but one thing is clear: because of his influence temporal logic was an essentially semantically driven enterprise. The best way into his work is via Prior [369].

With the work of Jónsson and Tarski [260, 261] we reach the most important (and puzzling) might-have-beens in the history of modal logic. Briefly, Jónsson and Tarski investigated the representation theory of boolean algebras with operators (that is, modal algebras). As we have remarked, while modal algebras were useful tools, they *seemed* of little help in guiding logical intuitions. The representation theory of Jónsson and Tarski should have swept this apparent shortcoming away for good, for in essence they showed how to represent modal algebras as the structures we now call models! In fact, they did a lot more than this. Their representation technique is essentially a model building technique, hence their work gave the technical tools needed to prove the completeness result that dominated the classical era (indeed, their approach is an algebraic analog of the canonical model technique that emerged 15 years later). Moreover, they provided all this for modal languages of arbitrary similarity type, not simply the basic modal language.

Unfortunately, their work was overlooked for 20 years; not until the start of the modern era was its significance appreciated. It is unclear to us why this happened. Certainly it didn't help matters that Jónsson and Tarski do not mention modal logic in their classic article; this is curious since Tarski had already published joint papers with McKinsey on algebraic approaches to modal logic. Maybe Tarski didn't see the connection at all: Copeland [94, page 13] writes that Tarski heard Kripke speak about relational semantics at a 1962 talk in Finland, a talk in which Kripke stressed the importance of the work by Jónsson and Tarski. According to Kripke, following the talk Tarski approached him and said he was unable to see any connection between the two lines of work.

Even if we admit that a connection which nows seems obvious may not have been so at the time, a puzzle remains. Tarski was based in California, which in the 1960s was the leading center of research in modal logic, yet in all those years, the connection was never made. For example, in 1966 Lemmon (also based in California) published a two part paper on algebraic approaches to modal logic [302] which reinvented (some of) the ideas in Jónsson and Tarski (Lemmon attributes these ideas to Dana Scott), but only cites the earlier Tarski and McKinsey papers.

We present the work by Jónsson and Tarski in Chapter 5; their Representation Theorem underpins the work of the entire chapter.

The classical era (1959–1972)

'Revolutionary' is an overused word, but no other word adequately describes the impact relational semantics (that is, the concepts of frames, models, satisfaction, and validity presented in this chapter) had on the study of modal logic. Problems which had previously been difficult (for example, distinguishing Hilbert-systems) suddenly yielded to straightforward semantic arguments. Moreover, like all revolutions worthy of the name, the new world view came bearing an ambitious research program. Much of this program revolved around the concept of completeness: at last is was possible to give a precise and natural meaning to claims that a logic generated everything it ought to. (For example, **K4** could now be claimed complete in a genuinely interesting sense: it generated *all* the formulas valid on transitive frames.) Such semantic characterizations are both simple and beautiful (especially when viewed against the complexities of the preceding era) and the hunt for such results was to dominate technical work for the next 15 years. The two outstanding monographs of the classical era — the existing fragment of Lemmon and Scott's *Intensional Logic* [303], and Segerberg's *An Essay in Classical Modal Logic* [396] — are largely devoted to completeness issues.

Some controversy attaches to the birth of the classical era. Briefly, relational semantics is often called Kripke semantics, and Kripke [290] (in which **S5**-based modal predicate logic is proved complete with respect to models with an implicit global relation), Kripke [291] (which introduces an explicit accessibility relation R and gives semantic characterization of some propositional modal logics in terms of this relation) and Kripke [292] (in which relational semantics for first-order modal languages is defined) were crucial in establishing the relational approach: they are clear, precise, and ever alert to the possibilities inherent in the new framework: for example, Kripke [292] discusses provability interpretations of propositional modal languages. Nonetheless, Hintikka had already made use of relational semantics to analyze the concept of belief and distinguish logics, and Hintikka's ideas played an important role in establishing the new paradigm in philosophical circles; see, for example, [230]. Furthermore, it has since emerged that Kanger, in a series of papers and monographs published in 1957, had introduced the basic idea of relational semantics for propositional and first-order modal logic; see, for example, Kanger [266, 267]. And a number of other authors (such as Arthur Prior, and Richard Montague [341]) had either published or spoken about similiar ideas earlier. Finally, the fact remains that Jónsson and Tarski had already presented and generalized the mathematical ideas needed to analyze propositional modal logics (though they do not discuss first-order modal languages).

But disputes over priority should not distract the reader from the essential point: somewhere around 1960 modal logic was reborn as a new field, acquiring new questions, methods, and perspectives. The magnitude of the shift, not who did what when, is what is important here. (The reader interested in more detail on who did what when, should consult Goldblatt [188]. Incidentally, after carefully considering the evidence, Goldblatt concludes that Kripke's contributions were the most significant.)

So by the early 1960s it was was clear that relational semantics was an important tool for classifying modal logics. But how could its potential be unlocked? The key tool required — the *canonical models* we discuss in Chapter 4 — emerged with surprising speed. They seem to have first been used in Makinson [314] and in Cresswell [97] (although Cresswell's so-called subordination relation differs slightly from the canonical relation), and in Lemmon and Scott [303] they appear full-fledged in the form that has become standard.

Lemmon and Scott [303] is a fragment of an ambitious monograph that was intended to cover all then current branches of modal logic. At the time of Lemmon's death in 1966, however, only the historical introduction and the chapter on the basic modal languages had been completed. Nonetheless, it's a gem. Although for the next decade it circulated only in manuscript form (it was not published until 1977) it was enormously influential, setting much of the agenda for subsequent developments. It unequivocally established the power of the canonical model technique, using it to prove general results of a sort not hitherto seen. It also introduced *filtrations*, an important technique for building finite models we will discuss in Chapter 2, and used them to prove a number of decidability results.

While Lemmon and Scott showed how to exploit canonical models directly, many important normal logics (notably, **KL** and the modal and temporal logic of structures such as $(\mathbb{N},<)$, $(\mathbb{Z},<)$, $(\mathbb{Q},<)$, and $(\mathbb{R},<)$, and their reflexive counterparts) cannot be analyzed in this way. However, as Segerberg [396, 395] showed, it is possible to use canonical models indirectly: one can transform the canonical model into the required form and prove these (and a great many other) completeness results. Segerberg-style transformation proofs are discussed in Section 4.5.

But although completeness and canonical models were the dominant issues of the classical era, there is a small body of work which anticipates more recent themes. For example, Robert Bull, swimming against the tide of fashion, used *algebraic* arguments to prove a striking result: all normal extensions of **S4.3** are characterized by classes of finite *models* (see Bull [72]). Although model-theoretic proofs of Bull's Theorem were sought (see, for example, Segerberg [396, page 170]), not until Fine [136] did these efforts succeed. Kit Fine was shortly to play a key role in the birth of the modern era, and the technical sophistication which was to characterize his later work is already evident in this paper; we discuss Fine's proof in Theorem 4.96. As a second example, in his 1968 PhD thesis [263], Hans Kamp proved one of the few (and certainly the most interesting) *expressivity* result of the era. He defined two natural binary modalities, since and until (discussed in Chapter 7), showed that the standard temporal language was not strong enough to define them, and proved that over Dedekind continuous strict total orders (such as $(\mathbb{R},<)$) his new modalities offered full first-order expressive power.

Summing up, the classical era supplied many of the fundamental concepts and methods used in contemporary modal logic. Nonetheless, viewed from a modern perspective, it is striking how differently these ideas were put to work then. For a start, the classical era took over many of the *goals* of the syntactic era. Modal investigations still revolved round much the same group of concepts: necessity, belief, obligation and time. Moreover, although modal research in the classical era was certainly not syntactical, it was, by and large, *syntactically driven*. That is — with the notable exception of the temporal tradition — relational semantics seems to have been largely viewed as a tool for analyzing logics: soundness results could distinguish logics, and completeness results could give them nice characterizations. Relational structures, in short, weren't really there to be *described* — they were there to fulfill an analytic role. (This goes a long way towards explaining the lack of expressivity results for the basic modal language; Kamp's result, significantly, was grounded in the Priorean tradition of temporal logic.) Moreover, it was a self-contained world in a way that modern modal logic is not. Modal languages and relational semantics: the connection between them seemed clear, adequate, and well understood. Surely nothing essential was missing from this paradise?

The modern era (1972–present)

Two forces gave rise to the modern era: the discovery of frame incompleteness results, and the adoption of modal languages in theoretical computer science. These unleashed a wealth of activity which profoundly changed the course of modal logic and continues to influence it till this day. The incompleteness results results forced a fundamental reappraisal of what modal languages actually *are*, while the influence of theoretical computer science radically changed expectations of *what* they could be used for, and *how* they were to be applied.

Frame-based analyses of modal logic were revealing and intoxicatingly successful — but was *every* normal logic complete with respect to some class of frames? Lemmon and Scott knew that this was a difficult question; they had shown, for example that there were obstacles to adapting the canonical model method to analyze the logic yielded by McKinsey axiom. Nonetheless, they conjectured that the answer was *yes*:

> However, it seems reasonable to conjecture that, if a consistent normal K-system S is *closed with respect to substitution instances* ... then S determines a class Γ_S of world systems such that $\vdash_S$ **A** iff $\models^{\Gamma_S}$ **A**. We have no proof of this conjecture. But to prove it would be to make a considerable difference to our theoretical understanding of the general situation. [303, page 76]

Other optimistic sentiments can be found in the literature of the period. Segerberg's thesis is more cautious, simply identifying it as 'probably the outstanding question in this area of modal logic at the present time' [396, page 29].

The question was soon resolved — *negatively*. In 1972, S.K. Thomason [426] showed that there were incomplete normal logics in the basic temporal language, and in 1974 Thomason [427] and Fine [137] both published examples of incomplete normal logics in the basic modal language. Moreover, in an important series of papers Thomason showed that these results were ineradicable: as tools for talking about frames, modal languages were essentially monadic second-order logic in disguise, and hence were intrinsically highly complex.

These results stimulated what remains some of the most interesting and innovative work in the history of the subject. For a start, it was now clear that it no longer sufficed to view modal logic as an isolated formal system; on the contrary, it was evident that a full understanding of what modal languages were, required that their position in the logical universe be located as accurately as possible. Over the next few years, modal languages were to be extensively mapped from the perspective of both *universal algebra* and *classical model theory*.

Thomason [426] had already adopted an algebraic perspective on the basic temporal language. Moreover, this paper introduced general frames, showed that they were equivalent to semantics based on boolean algebras with operators, and showed that these semantics were complete in a way that the frame-based semantics was not: every normal temporal logic was characterized by some algebra. Goldblatt introduced the universal algebraic approach towards modal logic and developed modal duality theory (the categorical study of the relation between relational structures endowed with topological structure on the one hand, and boolean algebras with operators on the other). This led to a belated appreciation of the fundamental contributions made in Jónsson and Tarski's pioneering work. Goldblatt and Thomason showed that the concepts and results of universal algebra could be applied to yield modally interesting results; the best known example of this is the Goldblatt-Thomason theorem a model theoretic characterization of modally definable frame classes obtained by applying the Birkhoff variety theorem to boolean algebras with operators. We discuss such work in Chapter 5 (and in Chapter 3 we discuss the Goldblatt-Thomason theorem from the perspective of first-order model theory). Work by Blok made deeper use of algebras, and universal algebra became a key tool in the exploration of completeness theory (we briefly discuss Blok's contribution in the Notes to Chapter 5). The revival of algebraic semantics — together with a genuine appreciation of *why* it was so important — is one of the most enduring legacies of this period.

But the modern period also firmly linked modal languages with classical model theory. One line of inquiry that led naturally in this direction was the following: given that modal logic was essentially second-order in nature, why was it so often first-order, and very simple first-order at that? That is, from the modern perspective, incomplete normal logics were to be expected — it was the elegant results of the classical period that now seemed in need of explanation. One type of answer was given in the work of Sahlqvist [388], who isolated a large set of axioms which guaranteed completeness with respect to first-order definable classes of frames. (We define the Sahlqvist fragment in Section 3.6, where we discuss the Sahlqvist Correspondence Theorem, an expressivity result. The twin Sahlqvist Completeness Theorem is proved algebraically in Theorem 5.91.) Another type of answer was developed in Fine [140] and van Benthem [39, 40]; we discuss this work (albeit from an algebraic perspective) in Chapter 5.

A different line of work also linked modal and classical languages: an investigation of modal languages viewed purely as *description languages*. As we have mentioned, the classical era largely ignored expressivity in favor of completeness. The Sahlqvist Correspondence Theorem showed the narrowness of this perspective: here was a beautiful result about the basic modal language that did not even mention normal modal logics! Expressivity issues were subsequently explored by van Benthem, who developed the subject now known as *correspondence theory*; see [41, 42]. His work has two main branches. One views modal languages as tools for describing *frames* (that is, as second-order description languages) and probes their expressive power. This line of investigation, together with Sahlqvist's pioneering work, forms the basis of Chapter 3. The second branch explores modal languages as tools for talking about *models*, an intrinsically first-order perspective. This lead van Benthem to isolate the concept of a *bisimulation*, and prove the fundamental Characterization Theorem: viewed as a tool for talking about models, modal languages are the bisimulation invariant fragment of the corresponding first-order language. Bisimulation driven investigations of modal expressivity are now standard, and much of the following chapter is devoted to such issues.

The impact of theoretical computer science was less dramatic than the discovery of the incompleteness results, but its influence has been equally profound. Burstall [80] already suggests using modal logic to reason about programs, but the birth of this line of work really dates from Pratt [367] (the paper which gave rise to PDL) and Pnueli [363] (which suggested using temporal logic to reason about execution-traces of programs). Computer scientists tended to develop powerful modal languages; PDL in its many variants is an obvious example (see Harel [215] for a detailed survey). Moreover, since the appearance of Gabbay *et al.* [167], the temporal languages used by computer scientists typically contain the until operator, and often additional operators which are evaluated with respect to *paths* (see Clarke and Emerson [92]). Gabbay also noted the significance of Rabin's theorem [372] for modal decidability (we discuss this in Chapter 6), and applied it to a wide range of languages and logics; see Gabbay [155, 156, 154].

Computer scientists brought a new array of questions to the study of modal logic. For a start, they initiated the study of the computational complexity of normal logics. Already by 1977 Ladner [299] had showed that every normal logic between **K** and **S4** had a PSPACE-hard satisfiability problem, while the results of Fischer and Ladner [143] and Pratt [366] together show that PDL has an EXPTIME-complete satisfiability problem. (These results are proved in Chapter 6.) Moreover, the interest of the modal expressivity studies emerging in correspondence theory was reinforced by several lines of work in computer science. To give one particularly nice example, computer scientists studying concurrent systems independently isolated the notion of bisimulation (see Park [358]). This paved the way for the work of Hennessy and Milner [225] who showed that weak modal languages could be used to classify various notions of process invariance.

But one of the most significant endowments from computer science has actually been something quite simple: it has helped remove a lingering tendency to see modal languages as intrinsically 'intensional' formalisms, suitable only for analyzing such concepts as knowledge, obligation and belief. During the 1990s this point was strongly emphasized when connections were discovered between modal logic and knowledge representation formalisms. In particular, *description logics* are a family of languages that come equipped with effective reasoning methods, and a special focus on balancing expressive power and computational and algorithmic complexity; see Donini *et al.* [123]. The discovery of this connection has lead to a renewed focus on efficient reasoning methods, dedicated languages that are fine-tuned for specific modeling tasks, and a variety of novel uses of modal languages; see Schild [392] for the first paper to make the connection between the two fields, and De Giacomo [102] and Areces [12, 15] for work exploiting the connection.

And this is but one example. Links with computer science and other disciplines have brought about an enormous richness and variety in modal languages. Computer science has seen a shift of emphasis from isolated programs to complex entities collaborating in heterogeneous environments; this gives rise to new challenges for the use of modal logic in theoretical computer science. For instance, agent-based theories require flexible modeling facilities together with efficient reasoning mechanisms; see Wooldridge and Jennings [455] for a discussion of the agent paradigm, and Bennet *et al.* [33] for the link with modal logic. More generally, complex computational architectures call for a variety of combinations of modal languages; see the proceedings of the *Frontiers of Combining Systems* workshop series for references [16, 160, 273].

Similar developments took place in foundational research in economics. Game theory (Osborne and Rubinstein [354]) also shows a nice interplay between the notions of action and knowledge; recent years have witnessed an increasing tendency to give a formal account of epistemic notions, cf. Battigalli and Bonanno [30] or Kaneko and Nagashima [265]. For modal logics that combine dynamic and epistemic notions to model games we refer to Baltag [20] and van Ditmarsch [117].

Further examples abound. Database theory continues to be a fruitful source of questions for logicians, modal or otherwise. For instance, developments in temporal databases have given rise to new challenges for temporal logicians (see Finger [142]), while decription logicians have found new applications for their modeling and reasoning methods in the area of semistructured data (see Calvanese *et al.* [82]). In the related, but more philosophically oriented area of belief revision, Fuhrmann [152] has given a modal formalization of one of the most influential approaches in the area, the AGM approach [4]. Authors such as Friedman and Halpern [150], Gerbrandy and Groeneveld [177], De Rijke [112], and Segerberg [403] have discussed various alternative modal formalizations.

Cognitive phenomena have long been of interest to modal logicians. This is clear from examples such as belief revision, but perhaps even more so from language-related work in modal logic. The feature logic mentioned in Example 1.17 is but one example; authors such as Blackburn, Gardent, Meyer Viol, and Spaan [59, 53], Kasper and Rounds [271, 386], Kurtonina [294], Kracht [287], and Reape [378] have offered a variety of modal logical perspectives on grammar formalisms, while others have analyzed the semantics of natural language by modal means; see Fernando [134] for a sample of modern work along these lines.

During the 1980s and 1990s a number of new themes on the interface of modal logic and mathematics received considerable attention. One of these themes concerns links between modal logic and non-wellfounded set theory; work that we should certainly mention here includes Aczel [2], Barwise and Moss [26], and Baltag [19, 21]; see the Notes to Chapter 2 for further discussion. Non-wellfounded sets and many other notions, such as automata and labeled transition systems, have been brought together under the umbrella of co-algebras (cf. Jacobs and Rutten [248]), which form a natural and elegant way to model state-based dynamic systems. Since it was discovered that modal logic is as closely related to co-algebras as equational logic is to algebras, there has been a wealth of results reporting on this connection; we only mention Jacobs [247], Kurz [297] and Rößiger [385] here.

Another 1990s theme on the interface of modal logic and mathematics concerns an old one: geometry. Work by Balbiani *et al.* [18], Stebletsova [416] and Venema [441] indicates that modal logic may have interesting things to say about geometry, while Aiello and van Benthem [3] and Lemon and Pratt [304] investigate the potential of modal logic as a tool for reasoning about space.

As should now be clear to all our readers, the simple question posed by the modal satisfaction definition — what happens at accessible states? — gives us a natural way of working with *any* relational structure. This has opened up a host of new applications for modal logic. Moreover, once the relational perspective has been fully assimilated, it opens up rich new approaches to traditional subjects: see van Benthem [44] and Fagin, Halpern, Moses, and Vardi [133] for thoroughly modern discussions of temporal logic and epistemic logic respectively.

1.8 Summary of Chapter 1

► *Relational Structures*: A relational structure is a set together with a collection of relations. Relational structures can be used to model key ideas from a wide range of disciplines.

- ▶ *Description Languages*: Modal languages are simple languages for describing relational structures.
- ▶ *Similarity Types*: The basic modal language contains a single primitive unary operator $\Diamond$. Modal languages of arbitrary similarity type may contain many modalities $\triangle$ of arbitrary arity.
- ▶ *Basic Temporal Language*: The basic temporal language has two operators F and P whose intended interpretations are 'at some time in the future' and 'at some time in the past.'
- ▶ *Propositional Dynamic Logic*: The language of propositional dynamic logic has an infinite collection of modal operators indexed by programs π built up from atomic programs using union $\cup$, composition ;, and iteration *; additional constructors such as intersection $\cap$ and test ? may also be used. The intended interpretation of $\langle\pi\rangle\phi$ is 'some terminating execution of program π leads to a state where ϕ holds.'
- ▶ *Arrow Logic*: The language of arrow logic is designed to talk about any object that may be represented by arrows; it has a modal constant 1' ('skip'), a unary operator $\otimes$ ('converse'), and a dyadic operator $\circ$ ('composition').
- ▶ *Satisfaction*: The *satisfaction definition* is used to interpret formulas inside models. This satisfaction definition has an obvious local flavor: modalities are interpreted as scanning the states accessible from the current state.
- ▶ *Validity*: A formula is *valid* on a frame when it is globally true, no matter what valuation is used. This concept allows modal languages to be viewed as languages for describing frames.
- ▶ *General Frames*: Modal languages can also be viewed as talking about general frames. A general frame is a frame together with a set of admissible valuations. General frames offer some of the advantages of both models and frames and are an important technical tool.
- ▶ *Semantic Consequence*: Semantic consequence relations for modal languages need to be relativized to classes of structures. The classical idea that the truth of the premises should guarantee the truth of the conclusion can be interpreted either locally or globally. In this book we almost exclusively use the local interpretation.
- ▶ *Normal Modal Logics*: Normal modal logics are the unifying concept in modal proof theory. Normal modal logics contain all tautologies, the K axiom and the Dual axiom; in addition they should be closed under modus ponens, uniform substitution and generalization.