IV

Modal Logic

1 Possibility and Necessity; Knowledge or Belief

Formal modal logics were developed to make precise the mathematical properties of differing conceptions of such notions as possibility, necessity, belief, knowledge and temporal progression which arise in philosophy and natural languages. In the last twenty–five years modal logics have emerged as useful tools for expressing essential ideas in computer science and artificial intelligence.

Formally, modal logic is an extension of classical propositional or predicate logic. The language of classical logic is enriched by adding of new "modal operators". The standard basic operators are traditionally denoted by $\Box$ and $\Diamond$. Syntactically, they can be viewed as new unary connectives. (We omit a separate treatment of propositional logic and move directly to predicate logic. As we noted for classical logic in II.4.8, propositional logic can be viewed as a subset of predicate logic and so is subsumed by it. The same translation works for modal logic.)

Definition 1.1: If $\mathcal{L}$ is a language for (classical) predicate logic (as defined in II.2), we extend it to a *modal language* $\mathcal{L}_{\Box,\Diamond}$ by adding (to Definition II.2.1) two new primitive symbols $\Box$ and $\Diamond$. We add a new clause to the definition (II.2.5) of formulas:

(iv) If φ is a formula, then so are $(\Box\varphi)$ and $(\Diamond\varphi)$.

The definitions of all other related notions such as subformula, bound variable and sentence are now carried over verbatim.

When no confusion is likely to result we drop the subscripts and refer to $\mathcal{L}_{\Box,\Diamond}$ as simply the (modal) language $\mathcal{L}$.

Interpretations of modal languages were originally motivated by philosophical considerations. Common readings of $\Box$ and $\Diamond$ are "it is necessary that" and "it is possible that". Another is "it will always be true that" and "it will eventually be true that". One should note that the intended relation between $\Box$ and $\Diamond$

A. Nerode et al., *Logic for Applications*

is like that between $\forall$ and $\exists$. They are dual operators in the sense that the intended meaning of $\Diamond\varphi$ is usually $\neg\Box\neg\varphi$. The two interpretations just mentioned have ordinary names for both operators. At times it is natural to use just one. Interpretations involving knowledge or belief, for example, are typically phrased in a language with just the operator $\Box$ (which could be denoted by $\mathcal{L}_\Box$) and $\Box\varphi$ is understood as "I know φ" or "I believe that φ". It is also possible to add on additional modal operators $\Box_i$ and $\Diamond_i$ and provide various interpretations. We prefer to read $\Box$ and $\Diamond$ simply as "box" and "diamond" so as not to prejudge the intended interpretation.

The semantics for a modal language $\mathcal{L}_{\Box,\Diamond}$ is based on a generalization of the structures for classical predicate logic of II.4 known as *Kripke frames*. Intuitively, we consider a collection W of "*possible worlds*". Each world $w \in W$ constitutes a view of reality as represented by a structure $\mathcal{C}(w)$ associated with it. We adopt the notation of forcing from set theory and write $w \Vdash \varphi$ to mean φ is true in the possible world w. (We read $w \Vdash \varphi$ as "w forces φ" or "φ is true at w".) If φ is a sentence of the classical language $\mathcal{L}$, this should be understood as simply asserting that φ is true in the structure $\mathcal{C}(w)$. If $\Box$ is interpreted as necessity, this notion can be understood as truth in all possible worlds; the notion of possibility expressed by $\Diamond$ would then mean truth in some possible world.

Temporal notions, or assertions of the necessity or possibility of some fact φ given some preexisting state of affairs, are expressed by including an accessibility (or successor) relation S between the possible worlds. Thus we write $w \Vdash \Box\varphi$ to mean that φ is true in all possible successor worlds of w or all worlds accessible from w. This is a reasonable interpretation of "φ is necessarily true in world w".

Before formalizing the semantics for modal logic in §2, we give some additional motivation by considering two types of applications to computer science.

The first area of application is to theories of program behavior. Modalities are implicit in the works of Turing [1949, 5.7], Von Neumann [1961, 5.7], Floyd [1967, 5.7], Hoare [1969, 5.7], and Burstall [1972, 5.7] on program correctness. The underlying systems of modal logic were brought to the surface by many later workers. Examples of the logics recently developed for the analysis of programs include algorithmic logic, dynamic logic, process logic and temporal logic. Here are the primitive modalities of one system, the dynamic logic of sequential programs.

Let α be a sequential (possibly nondeterministic) program and let s be a state of the machine executing α. Let φ be a predicate or property of states. We introduce modal operators $\Box_\alpha$ and $\Diamond_\alpha$ into the description of the execution of the program α with the intended interpretation of $\Box_\alpha\varphi$ being that φ is necessarily or always true after α is executed. The meaning of $\Diamond_\alpha$ is intended to be that φ is sometimes true when α is executed (i.e., there is some execution of α that makes φ true). Thus $\Box_\alpha$ is a modal necessity operator and $\Diamond_\alpha$ is a modal possibility operator.

We can make this language more useful by invoking the ideas of possible worlds as described above. Here the "possible worlds" are the states of the machine and the accessibility relation is determined by the possible execution sequences of the program α. More precisely, we interpret forcing assertions about modal formulas as follows:

> $s \Vdash \Box_\alpha \varphi$ asserts that φ is true at any state s' such that there exists a legal execution sequence for α which starts in state s and eventually reaches state s'.
> $s \Vdash \Diamond_\alpha \varphi$ asserts that φ is true at (at least) one state s' such that there exists a legal execution sequence for α which starts in state s and eventually reaches state s'.

Thus, the intended accessibility relation, S_α, is that s' is accessible from s, $sS_\alpha s'$, if and only if some execution of program α starting in state s ends in state s'.

We could just as well introduce separate operators $\Box_\alpha$, $\Diamond_\alpha$ for each program α. A modal Kripke semantics could then be developed with distinct accessibility relations S_α for each pair of operators $\Box_\alpha$ and $\Diamond_\alpha$. Such a language is very useful in discussing invariants of programs and, in general, proving their correctness. After all, correctness is simply the assertion that, no matter what the starting state, some situation φ is always true when the execution of α is finished: $\Box_\alpha \varphi$. (See, for example, Goldblatt [1982, 5.6], [1992, 5.6] and Harel [1979, 5.7].)

Many interesting and useful variations on this theme have been proposed. One could, for example, interpret $s \Vdash \Box_\alpha \varphi$ to mean that φ is true at every state s' that can be visited during an execution of α starting at s. In this vein, $s \Vdash \Diamond_\alpha \varphi$ would mean that φ is true at some state s' which is reached during some execution of α starting at s. We have simply changed the accessibility relation and we have what is called process logic. This interpretation is closely related to temporal logic. In temporal logic, $\Box\varphi$ means that φ is always true and $\Diamond\varphi$ means that φ will eventually (or at some time) be true. This logic can be augmented in various ways with other modal operators depending on one's view of time. In a digital sequential machine, it may be reasonable to view time as ordered as are the natural numbers. In this situation, for example, one can introduce a modal operator $\circ$ and read $t \Vdash \circ\varphi$ as φ is true at the moment which is the immediate successor of t. Various notions of fairness, for example, can be formulated in these systems (even without $\circ$): every constantly active process will eventually be scheduled (for execution or inspection etc.) — $\Box\varphi(c) \rightarrow \Diamond\psi(c)$; every process which is ever active is scheduled at least once — $\varphi(c) \rightarrow \Diamond\psi(c)$; every process active infinitely often will be scheduled infinitely often — $\Box\Diamond\varphi(c) \rightarrow \Box\Diamond\psi(c)$; etc. Thus these logics are relevant to analyses of the general behavior and, in particular, the correctness of concurrent or perpetual programs. (Another good reference here is Manna and Waldinger [1985, 5.6].)

A quite different source of applications of modal logic in computer science is in theories of knowledge and belief for AI. Here we may understand $\Box_K \varphi$ as "some (fixed) agent or processor knows φ (i.e., that φ is true)" or $\Box_B\varphi$ as "some (fixed) agent or processor believes φ (i.e., that φ is true)". Again, we may wish to discuss not one processor but many. We can then introduce modal operators such as $\Box_{K,\alpha}\varphi$ to be understood as "processor α knows φ". Thus, for example, $\Box_{K,\alpha}\Box_{K,\beta}\varphi$ says that α knows that β knows φ; $\Box_{K,\alpha}\varphi \rightarrow \Box_{K,\beta}\psi$ says that if α knows φ, then β knows ψ. (A general reference for AI logics is Turner [1984, 5.6].)

This language clearly allows one to formulate notions about communication and knowledge in distributed or concurrent systems. Another related avenue of investigation considers attempts to axiomatize belief or knowledge in humans as well as machine systems. One can then deduce what other properties of knowledge or belief follow from the axioms. On the basis of such deductions, one may either modify one's epistemological views or change the axioms about knowledge that one is willing to accept. The view of modal logic as a logic of belief or knowledge is particularly relevant to analyses of database management. In this light, it is also closely related to nonmonotonic logic as presented in III.7. (See Fagin et al. [1995, 5.6] for a survey of logics of knowledge and beliefs and Thayse [1989, 5.6] for a thorough treatment of modal logic aimed at deductive databases and AI applications.)

In the next sections we give a formal semantics for modal logic (§2) and a tableau style proof system (§3). In §4 we prove soundness and completeness theorems for our proof system. Many applications of modal logic concern systems in which there are agreed (or suggested) restrictions on the interpretations corresponding to varying views of the properties of necessity, knowledge, time, etc., that one is trying to capture. We devote §5 to the relation between restrictions on the accessibility relation, adding axioms about the modal operators to the underlying logic and adjoining new tableau proof rules. The final section (§6) describes a traditional Hilbert–style system for modal logic extending that presented for classical logic in II.8.

2 Frames and Forcing

For technical convenience, we make a couple of modifications to the basic notion of a (modal) language $\mathcal{L}$. First, we omit the connective $\leftrightarrow$ from our formal language and view $\varphi \leftrightarrow \psi$ as an abbreviation for $\varphi \rightarrow \psi \wedge \psi \rightarrow \varphi$. Second, we assume throughout this chapter that every language $\mathcal{L}$ has at least one constant symbol but no function symbols other than constants. (The elimination of function symbols does not result in a serious loss of expressiveness. We can systematically replace function symbols with relations. The work of a binary function symbol $f(x,y)$, for example, can be taken over by a ternary relation symbol $R_f(x,y,z)$ whose intended interpretation is that $f(x,y) = z$. A formula $\varphi(f(x,y))$ can then be systematically replaced by the formula $\exists z(R_f(x,y,z) \wedge \varphi(z))$.)

We now present the precise notion of a frame used to formalize the semantics of modal logic. As we have explained, a frame consists of a set W of "possible worlds", an accessibility (or successor) relation S between the possible worlds and an assignment of a classical structure $\mathcal{C}(p)$ to each $p \in W$. We have chosen to require that the domains $C(p)$ of the structures $\mathcal{C}(p)$ be monotonic in the successor relation, i.e., if q is a successor world of p, pSq, then $C(p) \subseteq C(q)$. This weak monotonicity requirement is not a serious restriction. As even the atomic predicates are not assumed to be monotonic, i.e. an element c of $C(p)$ can have some property R in $C(p)$ but not in $C(q)$, any object can be declared to no longer be in the domain of a particular database or other predicate. One can provide frame semantics that do not incorporate this restriction but there are many difficulties involved that we wish to avoid. For example, if all objects cease to exist, i.e., some $C(q) = \emptyset$, we have entirely left the realm of classical predicate logic which is formulated only for nonempty domains.

Definition 2.1: Let $C = (W, S, \{C(p)\}_{p \in W})$ consist of a set W, a binary relation S on W and a function that assigns to each p in W a (classical) structure $\mathcal{C}(p)$ for $\mathcal{L}$ (in the sense of Definition II.4.1). To simplify the notation we write $\mathcal{C} = (W, S, \mathcal{C}(p))$ instead of the more formally precise version, $\mathcal{C} = (W, S, \{C(p)\}_{p \in W})$. As usual, we let $C(p)$ denote the domain of the structure $\mathcal{C}(p)$. We also let $\mathcal{L}(p)$ denote the extension of $\mathcal{L}$ gotten by adding on a name c_a for each element a of $C(p)$ in the style of the definition of truth in II.4. We write either pSq or $(p, q) \in S$ to denote the fact that the relation S holds between p and q. We also describe this state by saying that q is *accessible* from (or a *successor* of) p. We say that $\mathcal{C}$ is a *frame* for the language $\mathcal{L}$, or simply an $\mathcal{L}$*-frame* if, for every p and q in W, pSq implies that $C(p) \subseteq C(q)$ and the interpretations of the constants in $\mathcal{L}(p) \subseteq \mathcal{L}(q)$ are the same in $\mathcal{C}(p)$ as in $\mathcal{C}(q)$.

We now define the forcing relation for $\mathcal{L}$–frames. While reading the definition and working through the later examples, it may help to keep in mind the following paradigm interpretation: each $p \in W$ is a possible world; pSq means that q is a possible future of p; $p \Vdash \varphi$ means that φ is true in the world p; $\Box\varphi$ means that φ will always be true and $\Diamond\varphi$ means that φ will be true sometime in the future.

Definition 2.2 (*Forcing for frames*)**:** Let $\mathcal{C} = (W, S, \mathcal{C}(p))$ be a frame for a language $\mathcal{L}$, p be in W and φ be a sentence of the language $\mathcal{L}(p)$. We give a definition of *p forces* φ, written $p \Vdash \varphi$ by induction on sentences φ.

(i) For atomic sentences φ, $p \Vdash \varphi \Leftrightarrow \varphi$ is true in $\mathcal{C}(p)$.

(ii) $p \Vdash (\varphi \to \psi) \;\Leftrightarrow\; p \Vdash \varphi$ implies $p \Vdash \psi$.

(iii) $p \Vdash \neg\varphi \;\Leftrightarrow\; p$ does not force φ (written $p \nVdash \varphi$).

(iv) $p \Vdash (\forall x)\varphi(x) \;\Leftrightarrow\;$ for every constant c in $\mathcal{L}(p)$, $p \Vdash \varphi(c)$.

(v) $p \Vdash (\exists x)\varphi(x) \;\Leftrightarrow\;$ there is a constant c in $\mathcal{L}(p)$ such that $p \Vdash \varphi(c)$.

(vi) $p \Vdash (\varphi \wedge \psi) \;\Leftrightarrow\; p \Vdash \varphi$ and $p \Vdash \psi$.

(vii) $p \Vdash (\varphi \vee \psi) \;\Leftrightarrow\; p \Vdash \varphi$ or $p \Vdash \psi$.

(viii) $p \Vdash \Box\varphi \;\Leftrightarrow\;$ for all $q \in W$ such that pSq, $q \Vdash \varphi$.

(ix) $p \Vdash \Diamond\varphi \;\Leftrightarrow\;$ there is a $q \in W$ such that pSq and $q \Vdash \varphi$.

If we need to make the frame explicit, we say that *p forces φ in $\mathcal{C}$* and write $p \Vdash_{\mathcal{C}} \varphi$.

Definition 2.3: Let φ be a sentence of the language $\mathcal{L}$. We say that *φ is forced in the $\mathcal{L}$-frame $\mathcal{C}$*, $\Vdash_{\mathcal{C}} \varphi$, if every p in W forces φ. We say φ is *valid*, $\models \varphi$, if φ is forced in every $\mathcal{L}$-frame $\mathcal{C}$.

Example 2.4: For any sentence φ, the sentence $\Box\varphi \rightarrow \neg\Diamond\neg\varphi$ is valid: Consider any frame $\mathcal{C} = (W, S, \mathcal{C}(p))$ and any $p \in W$. We must verify that $p \Vdash \Box\varphi \rightarrow \neg\Diamond\neg\varphi$ in accordance with Clause (ii) of Definition 2.2. Suppose then that $p \Vdash \Box\varphi$. If $p \nVdash \neg\Diamond\neg\varphi$, then $p \Vdash \Diamond\neg\varphi$ (by (iii)). By Clause (ix), there is a $q \in W$ such that pSq and $q \Vdash \neg\varphi$. Our assumption that $p \Vdash \Box\varphi$ and Clause (viii) then tell us that $q \Vdash \varphi$, contradicting Clause (iii). Exercise 1 shows that the converse, $\neg\Diamond\neg\varphi \rightarrow \Box\varphi$, is also valid.

Example 2.5: We claim that $\Box\forall x\varphi(x) \rightarrow \forall x\Box\varphi(x)$ is valid. If not, there is a frame $\mathcal{C}$ and a p such that $p \Vdash \Box\forall x\varphi(x)$ but $p \nVdash \forall x\Box\varphi(x)$. If $p \nVdash \forall x\Box\varphi(x)$, there is, by Clause (iv), a $c \in \mathcal{L}(p)$ such that $p \nVdash \Box\varphi(c)$. There is then, by Clause (ix), a $q \in W$ such that pSq and $q \nVdash \varphi(c)$. As $p \Vdash \Box\forall x\varphi(x)$, $q \Vdash \forall x\varphi(x)$ by (ix). Finally, $q \Vdash \varphi(c)$ by (iv) for the desired contradiction. Note that the assumption that the domains $C(p)$ are monotonic, in the sense that $pSq \Rightarrow C(p) \subseteq C(q)$, plays a key role in this argument.

Example 2.6: $\Box\varphi(c) \rightarrow \varphi(c)$ is not valid: Consider any frame in which the atomic sentence $\varphi(c)$ is not true in some $\mathcal{C}(p)$ and there is no q such that pSq. In such a frame $p \Vdash \Box\varphi(c)$ but $p \nVdash \varphi(c)$.

Example 2.7: $\forall x\varphi(x) \rightarrow \Box\forall x\varphi(x)$ is not valid: Let $\mathcal{C}$ be the frame in which $W = \{p, q\}$, $S = \{(p, q)\}$, $C(p) = \{c\}$, $C(q) = \{c, d\}$, $\mathcal{C}(p) \models \varphi(c)$ and $\mathcal{C}(q) \models \varphi(c) \wedge \neg\varphi(d)$. Now $p \Vdash \forall x\varphi(x)$ but $p \nVdash \Box\forall x\varphi(x)$ as $q \nVdash \varphi(d)$. It is crucial in this example that the domains $C(p)$ of a frame $\mathcal{C}$ are not assumed to all be the same. Modal logic restricted to constant domains is considered in Exercise 4.8.

Note that validity as defined here coincides with that for classical predicate logic for sentences φ with no modal operators (Exercise 10).

Some care must be taken now in the definition of "logical consequence" for modal logic. If one keeps in mind that the basic structure is the entire frame and not the individual worlds within it, one is led to the following definition:

Definition 2.8: Let Σ be a set of sentences in a modal language $\mathcal{L}$ and φ a single sentence of $\mathcal{L}$. φ *is a logical consequence of* Σ, $\Sigma \models \varphi$, if φ is forced in every $\mathcal{L}$ frame $\mathcal{C}$ in which every $\psi \in \Sigma$ is forced.

Warning: This notion of logical consequence is not the same as requiring that, in every $\mathcal{L}$-frame $\mathcal{C}$, φ is true (forced) at every world w at which every $\psi \in \Sigma$ is forced (Exercise 11). In particular, the deduction theorem (Exercise II.7.6) fails for modal logic as can be seen from Examples 2.7 and 2.9.

Example 2.9: $\forall x\varphi(x) \models \Box\forall x\varphi(x)$: Suppose $\mathcal{C}$ is a frame in which $p \Vdash \forall x\varphi(x)$ for every possible world $p \in W$. If $q \in W$, we claim that $q \Vdash \Box\forall x\varphi(x)$. If not, there would be a $p \in W$ such that qSp and $p \nVdash \forall x\varphi(x)$ contradicting our assumption.

Example 2.10: If φ is an atomic unary predicate, $\Box\varphi(c) \not\models \Diamond\varphi(c)$: Consider a frame $\mathcal{C}$ in which $S = \emptyset$ and in which $\mathcal{C}(p) \not\models \varphi(c)$ and so $p \nVdash \varphi(c)$ for every p. In $\mathcal{C}$, every p forces $\Box\varphi(c)$ but none forces $\varphi(c)$ and so none forces $\Diamond\varphi(c)$.

There are other notions of validity (and so of logical consequence) that result from putting further restrictions on the set W of possible worlds or (more frequently) on the accessibility relation S. For example, it is often useful to consider only reflexive and transitive accessibility relations. We discuss several such alternatives in §5.

One should be aware that although $\Box$ and $\Diamond$ are treated syntactically as propositional connectives, their semantics involves quantification over all possible accessible worlds. $\Box\varphi$ says that, no matter what successor world one might move to, φ will be true there. $\Diamond\varphi$ says that there is some successor world to which one could move and make φ true. The construction of tableaux appropriate to such semantics will involve, of course, the introduction of new worlds and instantiations for elements of old ones.

Exercises

Prove, on the basis of the semantic definition of validity in Definition 2.3, that the following are valid modal sentences.

1. $\neg\Diamond\neg\varphi \rightarrow \Box\varphi$ (for any sentence φ).
2. $\forall x\Box\varphi(x) \rightarrow \exists x\Box\varphi(x)$ (for any formula $\varphi(x)$ with only x free).

Prove that the following are not, in general, valid modal sentences. Let φ be any modal sentence.

3. $\varphi \to \Diamond\varphi$.
4. $\varphi \to \Box\varphi$.
5. $\Diamond\varphi \to \varphi$.

Verify the following instances of logical consequence for modal sentences φ:

6. $\varphi \models \Box\varphi$.
7. $(\varphi \to \Box\varphi) \models (\Box\varphi \to \Box\Box\varphi)$.

Give frames that demonstrate the following failures of logical consequence:

8. $\Box\varphi \not\models \varphi$.
9. $(\Box\varphi \to \varphi) \not\models (\Box\varphi \to \Box\Box\varphi)$.
10. If φ is a sentence with no occurrences of $\Box$ or $\Diamond$, prove that validity for φ in the sense of Definition 3.1 coincides with that of II.4.4.
11. We say that φ *is a local consequence of* Σ if, for every $\mathcal{L}$-frame $\mathcal{C} = (W, S, \mathcal{C}(p))$, $\forall p \in W[(\forall\psi \in \Sigma)(p \Vdash \psi) \to p \Vdash \varphi]$.
    (i) Prove that if φ is a local consequence of Σ, then it is a logical consequence of Σ.
    (ii) Prove that the converse of (i) fails, i.e., φ may be a logical consequence of Σ without being a local consequence.

3 Modal Tableaux

We describe a proof procedure for modal logic based on a tableau-style system like that used for classical logic in II.6. In classical logic, the plan guiding tableau proofs is to systematically search for a structure agreeing with the starting signed sentence. We either get such a structure or see that each possible analysis leads to a contradiction. When we begin with a signed sentence $F\varphi$, we thus either find a structure in which φ fails or decide that we have a proof of φ. For modal logic we instead begin with a *signed forcing assertion* $Tp \Vdash \varphi$ or $Fp \Vdash \varphi$ (φ is again a sentence) and try either to build a frame agreeing with the assertion or decide that any such attempt leads to a contradiction. If we begin with $Fp \Vdash \varphi$, we either find a frame in which p does not force φ or decide that we have a modal proof of φ.

The definitions of tableau and tableau proof for modal logic are formally very much like those of II.6 for classical logic. *Modal tableaux* and *tableau proofs* are labeled binary trees. The labels (again called the *entries of the tableau*) are now either *signed forcing assertions* (i.e., labels of the form $Tp \Vdash \varphi$ or $Fq \Vdash \varphi$ for φ a sentence of any given appropriate language) or accessibility assertions pSq. We read $Tp \Vdash \varphi$ as p forces φ and $Fp \Vdash \varphi$ as p does not force φ.

As we are using ordinary predicate logic within each possible world, the atomic tableaux for the propositional connectives $\vee, \wedge, \neg$ and $\rightarrow$ are as in the classical treatment in I.4 or II.6 except that their entries are now signed forcing assertions. The atomic tableaux for the quantifiers $\forall$ and $\exists$ are designed to reflect both the previous concerns in predicate logic as well as our monotonicity assumptions about the domains of possible worlds under the accessibility relation. Thus we still require that only "new" constants be used as witnesses for a true existential sentence or as counterexamples to false universal ones. Roughly speaking, a "new" constant is one for which no previous commitments have been made, e.g., one not in $\mathcal{L}$ or appearing so far in the tableau. Consider, on the other hand, a true universal sentence, $Tp \Vdash \forall x\varphi(x)$. In classical predicate logic we could substitute any constant at all for the universally quantified variable x. Here we can conclude $Tp \Vdash \varphi(c)$ only for constants c which we know to be in $C(p)$ or in $C(q)$ for some world q from which p is accessible, qSp. This idea translates into the requirement that c is in $\mathcal{L}$ or has appeared in a forcing assertion on the path so far that involves p or some q for which qSp has also appeared on the path so far. The point here is that, if qSp and c is in $C(q)$, then by monotonicity it must be in $C(p)$ as well. In the description of modal tableaux, we refer to these constants as "any appropriate c". Of course, the formal definitions of both "new" and "appropriate" constants are given along with the definition of tableaux.

The other crucial element is the treatment of signed forcing sentences beginning with $\Box$ or $\Diamond$. In classical logic, the elements of the structure built by developing a tableau were the constant symbols appearing on some path of the tableau. We are now attempting to build an entire frame. The p's and q's appearing in the entries of some path P through our tableau constitute the possible worlds of the frame. We must also specify some appropriate accessibility relation S along each path of the tableau. It is convenient to include this information directly on the path. Thus we allow as entries in the tableau facts of the form pSq for possible worlds p and q that appear in signed forcing assertions on the path up to the entry. Entries of this form are put on the tableau by some of the atomic tableaux for $\Box$ and $\Diamond$. For example, from $Tp \Vdash \Diamond\varphi$ we can (semantically) conclude that $Tq \Vdash \varphi$ for some q such that pSq. Thus the atomic tableau for $Tp \Vdash \Diamond\varphi$ puts both pSq and $Tq \Vdash \varphi$ on the path for some new q (i.e., one not appearing in the tableau so far). On the other hand, the atomic tableau for $Tp \Vdash \Box\varphi$ reflects the idea that the meaning of $p \Vdash \Box\varphi$ is that φ is true in every world q such that pSq. It puts on the path the assertion $Tq \Vdash \varphi$ for any appropriate q, i.e., any q for which we already know that pSq by virtue of the fact that pSq has itself appeared on the path so far. In this way, we are attempting to build a suitable frame along every path of the tableau.

We now formally specify the *atomic tableaux*.

Definition 3.1 (*Atomic tableaux*): We begin by fixing a modal language $\mathcal{L}$ and an expansion to $\mathcal{L}_C$ given by adding new constant symbols c_i for $i \in \mathcal{N}$. We list in Figure 43 the atomic tableaux (for the language $\mathcal{L}$). In the tableaux in the following list, φ and ψ, if unquantified, are any sentences in the language $\mathcal{L}_C$. If quantified, they are formulas in which only x is free.

Warning: In (T$\Box$) and (F$\Diamond$) we allow for the possibility that there is no appropriate q by admitting $Tp \Vdash \Box\varphi$ and $Fp \Vdash \Diamond\varphi$ as instances of (T$\Box$) and (F$\Diamond$), respectively.

The formal definition of tableaux is now quite similar to that for classical logic in II.3.

Definition 3.2: We continue to use our fixed modal language $\mathcal{L}$ and its extension by constants $\mathcal{L}_C$. We also fix a set $\{p_i | \ i \in \mathcal{N}\}$ of potential candidates for the p's and q's in our forcing assertions. A *modal tableau* (for $\mathcal{L}$) is a binary tree labeled with signed forcing assertions or accessibility assertions; both sorts of labels are called *entries* of the tableau. The class of modal tableaux (for $\mathcal{L}$) is defined inductively as follows.

i) Each atomic tableau τ is a tableau. The requirement that c be new in cases (T$\exists$) and (F$\forall$) here simply means that c is one of the constants c_i added on to $\mathcal{L}$ to get $\mathcal{L}_C$ which does not appear in φ. The phrase "any appropriate c" in (F$\exists$) and (T$\forall$) means any constant in $\mathcal{L}$ or in φ. The requirement that q be new in (F$\Box$) and (T$\Diamond$) here means that q is any of the p_i other than p. The phrase "any appropriate q" in (T$\Box$) and (F$\Diamond$) in this case simply means that the tableau is just $Tp \Vdash \Box\varphi$ or $Fp \Vdash \Diamond\varphi$ as there is no appropriate q.

ii) If τ is a finite tableau, P a path on τ, E an entry of τ occurring on P and τ' is obtained from τ by adjoining an atomic tableau with root entry E to τ at the end of the path P, then τ' is also a tableau.

   The requirement that c be new in cases (T$\exists$) and (F$\forall$) here means that it is one of the c_i (and so not in $\mathcal{L}$) that do not appear in any entry on τ. The phrase "any appropriate c" in (F$\exists$) and (T$\forall$) here means any c in $\mathcal{L}$ or appearing in an entry on P of the form $Tq \Vdash \psi$ or $Fq \Vdash \psi$ such that qSp also appears on P.

   In (F$\Box$) and (T$\Diamond$) the requirement that q be new means that we choose a p_i not appearing in τ as q. The phrase "any appropriate q" in (T$\Box$) and (F$\Diamond$) means we can choose any q such that pSq is an entry on P.

iii) If $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ is a sequence of finite tableaux such that, for every $n \geq 0$, τ_{n+1} is constructed from τ_n by an application of (ii), then $\tau = \cup\tau_n$ is also a tableau.

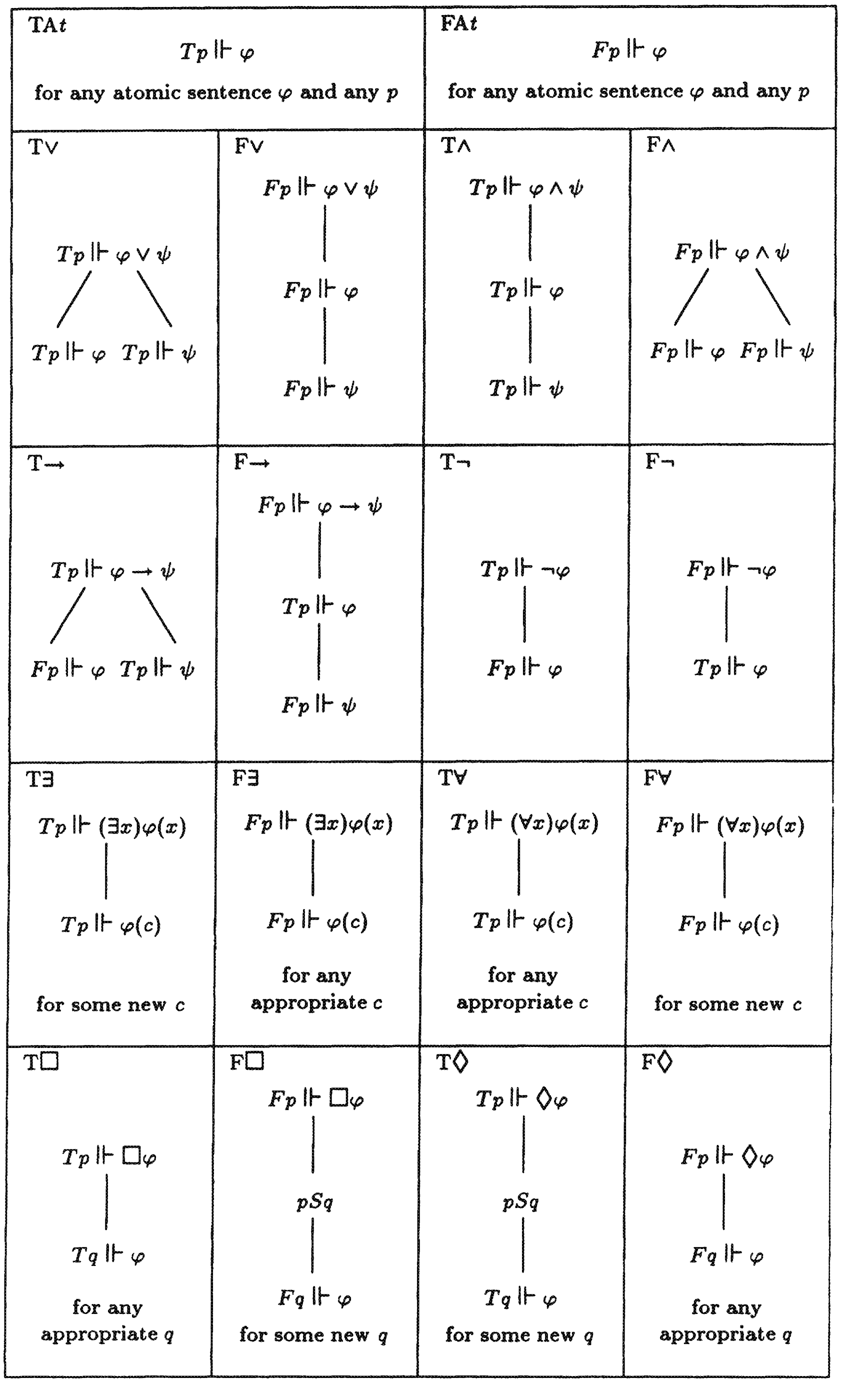
TAt
$Tp \Vdash \varphi$
for any atomic sentence φ and any p
FAt
$Fp \Vdash \varphi$
for any atomic sentence φ and any p
T∨
$Tp \Vdash \varphi \vee \psi$
$Tp \Vdash \varphi$ $Tp \Vdash \psi$
F∨
$Fp \Vdash \varphi \vee \psi$
$Fp \Vdash \varphi$
$Fp \Vdash \psi$
T∧
$Tp \Vdash \varphi \wedge \psi$
$Tp \Vdash \varphi$
$Tp \Vdash \psi$
F∧
$Fp \Vdash \varphi \wedge \psi$
$Fp \Vdash \varphi$ $Fp \Vdash \psi$
T→
$Tp \Vdash \varphi \to \psi$
$Fp \Vdash \varphi$ $Tp \Vdash \psi$
F→
$Fp \Vdash \varphi \to \psi$
$Tp \Vdash \varphi$
$Fp \Vdash \psi$
T¬
$Tp \Vdash \neg\varphi$
$Fp \Vdash \varphi$
F¬
$Fp \Vdash \neg\varphi$
$Tp \Vdash \varphi$
T∃
$Tp \Vdash (\exists x)\varphi(x)$
$Tp \Vdash \varphi(c)$
for some new c
F∃
$Fp \Vdash (\exists x)\varphi(x)$
$Fp \Vdash \varphi(c)$
for any appropriate c
T∀
$Tp \Vdash (\forall x)\varphi(x)$
$Tp \Vdash \varphi(c)$
for any appropriate c
F∀
$Fp \Vdash (\forall x)\varphi(x)$
$Fp \Vdash \varphi(c)$
for some new c
T□
$Tp \Vdash \Box\varphi$
$Tq \Vdash \varphi$
for any appropriate q
F□
$Fp \Vdash \Box\varphi$
pSq
$Fq \Vdash \varphi$
for some new q
T◊
$Tp \Vdash \Diamond\varphi$
pSq
$Tq \Vdash \varphi$
for some new q
F◊
$Fp \Vdash \Diamond\varphi$
$Fq \Vdash \varphi$
for any appropriate q

FIGURE 43.

As in the previous definitions, we insist that the entry E in Clause (ii) formally be repeated when the corresponding atomic tableau is added on to P to guarantee the property corresponding to a classical tableau being finished. The atomic tableaux for which they are actually needed are (F$\exists$), (T$\forall$), (T$\Box$) and (F$\Diamond$). However, we generally omit the repetition of the root entry in our examples as a notational convenience. The definition of tableau proofs now follows the familiar pattern.

Definition 3.3 (Tableau Proofs): Let τ be a modal tableau and P a path in τ.

i) P is *contradictory* if, for some forcing assertion $p \Vdash \varphi$, both $Tp \Vdash \varphi$ and $Fp \Vdash \varphi$ appear as entries on P.

ii) τ is *contradictory* if every path through τ is contradictory.

iii) τ is a *proof of* φ if τ is a finite contradictory modal tableau with its root node labeled $Fp \Vdash \varphi$ for some p. φ is *provable*, $\vdash \varphi$, if there is a proof of φ.

Note that, as in classical logic, if there is any contradictory tableau with root node $Fp \Vdash \varphi$, then there is one that is finite, i.e., a proof of φ : Just terminate each path when it becomes contradictory. As each path is now finite, the whole tree is finite by König's lemma. Thus, the added requirement that proofs be finite (tableaux) has no effect on the existence of proofs for any sentence. Another point of view is that we could have required that the path P in Clause (ii) of the definition of tableaux be noncontradictory without affecting the existence of proofs. Thus, in practice, when attempting to construct proofs we mark any contradictory path with the symbol $\otimes$ and terminate the development of the tableau along that path.

Before dealing with the soundness and completeness of the tableau method for modal logic, we look at some examples of modal tableau proofs. Remember that we are abbreviating the tableaux by generally not repeating the entry that we are expanding. We also number the levels of the tableau on the left and indicate on the right the level of the atomic tableau whose development produced the line.

Example 3.4: There is a natural correspondence between the tableaux of classical predicate logic and those of modal logic beginning with sentences without modal operators. One goes from the modal tableau to the classical one by replacing signed forcing assertions $Tp \Vdash \varphi$ and $Fp \Vdash \varphi$ by the corresponding signed sentences $T\varphi$ and $F\varphi$, respectively. (Formally one must account for the alternate notion of new constant used in II.6.1 when going in the other direction.) Note that this correspondence takes proofs to proofs. (See Exercise 1.)

Example 3.5: $\varphi \to \Box\varphi$, sometimes called the *scheme of necessitation*, is not valid. Figure 44 gives an attempt at a tableau proof.

1	$Fw \Vdash \varphi \to \Box\varphi$	
2	$Tw \Vdash \varphi$	by 1
3	$Fw \Vdash \Box\varphi$	by 1
4	wSv for a new v	by 3
5	$Fv \Vdash \varphi$	by 3

FIGURE 44.

This failed attempt at a proof suggests a frame counterexample $\mathcal{C}$ for which $W = \{w, v\}$, $S = \{(w, v)\}$ and structures such that φ is true at w but not at v. Such a frame demonstrates that $\varphi \to \Box\varphi$ is not valid as in this frame, w does not force $\varphi \to \Box\varphi$.

Example 3.6: Similarly $\Box\varphi \to \varphi$ is not valid as can be seen from the attempted proof in Figure 45.

1	$Fw \Vdash \Box\varphi \to \varphi$	
2	$Tw \Vdash \Box\varphi$	by 1
3	$Fw \Vdash \varphi$	by 1

FIGURE 45.

The frame counterexample suggested here consists of a one world $W = \{w\}$ with empty accessibility relation S and φ false at w. It shows that $\Box\varphi \to \varphi$ is not valid.

Various interpretations of $\Box$ might tempt one to think that $\Box\varphi \to \varphi$ should be valid. For example, probably all philosophers would agree that if φ is necessarily true, it in fact is true. On the other hand, most but perhaps not all epistemologists would argue that if I know φ, it must also be true. Finally, few people would claim that (for any φ) if I believe φ, then φ is true. $\Box\varphi \to \varphi$ is traditionally called "T" or the "knowledge axiom". Under many interpretations of $\Box$, it should be valid. A glance at the attempted proof above shows us that, if we knew that wSw, we could quickly get the desired contradiction. Thus, there is a relation between T and the assumption that the accessibility relation is reflexive. In fact, not only is T valid in all frames with reflexive accessibility relations, but conversely any sentence valid in all such frames can be deduced from T. We make this correspondence and others like it precise in §5.

Example 3.7: We show in Figure 46 that $\Box(\forall x)\varphi(x) \to (\forall x)\Box\varphi(x)$ is provable.

1	$Fw \Vdash \Box(\forall x)\varphi(x) \to (\forall x)\Box\varphi(x)$	
2	$Tw \Vdash \Box(\forall x)\varphi(x)$	by 1
3	$Fw \Vdash (\forall x)\Box\varphi(x)$	by 1
4	$Fw \Vdash \Box\varphi(c)$	by 3
5	wSv	by 4
6	$Fv \Vdash \varphi(c)$	by 4
7	$Tv \Vdash (\forall x)\varphi(x)$	by 2, 5
8	$Tv \Vdash \varphi(c)$	by 7
	$\otimes$	by 6, 8

FIGURE 46.

Note the use of monotonicity in the derivation of lines 6 and 8 corresponding to the semantic argument in Example 2.5.

Example 3.8: Figure 47 gives a tableau proof of

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

1	$Fw \Vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$		
2	$Tw \Vdash \Box(\varphi \to \psi)$		by 1
3	$Fw \Vdash \Box\varphi \to \Box\psi$		by 1
4	$Tw \Vdash \Box\varphi$		by 3
5	$Fw \Vdash \Box\psi$		by 3
6	wSv new v		by 5
7	$Fv \Vdash \psi$		by 5
8	$Tv \Vdash \varphi$		by 4, 6
9	$Tv \Vdash \varphi \to \psi$		by 2, 6
10	$Fv \Vdash \varphi$	$Tv \Vdash \psi$	by 9
11	$\otimes$	$\otimes$	by 8, 7

FIGURE 47.

(The scheme illustrated in Figure 47 plays an important role in the Hilbert–style systems of modal logic presented in §6.)

Example 3.9: Figure 48 gives an incorrect proof of $\forall x\Box\varphi(x) \rightarrow \Box\forall x\varphi(x)$.

1	$Fw \Vdash (\forall x)\Box\varphi(x) \rightarrow \Box(\forall x)\varphi(x)$		
2	$Tw \Vdash (\forall x)\Box\varphi(x)$		by 1
3	$Fw \Vdash \Box(\forall x)\varphi(x)$		by 1
4	wSv		by 3
5	$Fv \Vdash (\forall x)\varphi(x)$		by 3
6	$Fv \Vdash \varphi(c)$	new c	by 5
7	$Tw \Vdash \Box\varphi(c)$		by 2
8	$Tv \Vdash \varphi(c)$		by 7
	$\otimes$		

FIGURE 48.

The false step occurs at line 7. On the basis of line 6 we can use c for instantiations in forcing assertions about v or any world accessible from v but we have no basis to use it in assertions about w. As in Example 2.7, such a move would be appropriate for an analysis of constant domain frames. (See Exercise 4.8.)

Example 3.10: $(\forall x)\neg\Box\varphi \rightarrow \neg\Box(\exists x)\varphi$ is not valid. The tableau in Figure 49 that begins with this formula is not a proof. It does, however, suggest how to construct a frame counterexample. Let it have the constant domain $C = \{c, d\}$; two worlds w and v with v accessible from w; no atomic sentences true at w and the sentence $\varphi(d)$ true at v. This frame provides the required counterexample.

1	$Fw \Vdash (\forall x)\neg\Box\varphi \to \neg\Box(\exists x)\varphi$		
2	$Tw \Vdash (\forall x)\neg\Box\varphi$		by 1
3	$Fw \Vdash \neg\Box(\exists x)\varphi$		by 1
4	$Tw \Vdash \Box(\exists x)\varphi$		by 3
5	$Tw \Vdash \neg\Box\varphi(c)$		by 2
6	$Fw \Vdash \Box\varphi(c)$		by 5
7	wSv		by 6
8	$Fv \Vdash \varphi(c)$		by 6
9	$Tv \Vdash (\exists x)\varphi$		by 4, 7
10	$Tv \Vdash \varphi(d)$	new d	by 9

FIGURE 49.

As with the semantic definition of logical consequence, one must take care in defining the notion of a modal tableau proof from a set Σ of sentences (which we often call premises). We must match the intuition that we are restricting our attention to frames in which the premises are forced. To do this, we allow the insertion in the tableau of entries of the form $Tp \Vdash \varphi$ for any appropriate possible world p and any $\varphi \in \Sigma$.

Definition 3.11: The definition of *modal tableaux from* Σ, a set of sentences of a modal language called premises, is the same as for simple modal tableaux in Definition 3.2 except that we allow one additional formation rule:

(ii′) If τ is a finite tableau from Σ, $\varphi \in \Sigma$, P a path in τ and p a possible world appearing in some signed forcing assertion on P, then appending $Tp \Vdash \varphi$

to the end of P produces a tableau τ' from Σ.

The notions used to define a tableau proof are now carried over from Definition 3.3 to ***tableau proofs from*** Σ by simply replacing "tableau" by "tableau from Σ". We write $\Sigma \vdash \varphi$ to denote that φ ***is provable from*** Σ, i.e., there is a proof of φ from Σ.

Example 3.12: Figure 50 gives a tableau proof of $\Box\forall x\varphi(x)$ from the premise $\forall x\varphi(x)$.

1	$Fp \Vdash \Box(\forall x)\varphi(x)$		
2	pSq		by 1
3	$Fq \Vdash (\forall x)\varphi(x)$		by 1
4	$Fq \Vdash \varphi(c)$	new c	by 3
5	$Tq \Vdash (\forall x)\varphi(x)$		premise
6	$Tq \Vdash \varphi(c)$		by 5
	$\otimes$		

FIGURE 50.

Exercises

1. Make precise the correspondence described in Example 3.4 and show that it takes tableau proofs in classical predicate logic to ones in modal logic. Conversely, if τ is a modal tableau proof of a sentence φ of classical logic, describe the appropriate transformation in the other direction and show that it takes τ to a classical proof of φ.

In Exercises 2–8, let φ and ψ be any formulas with either no free variables or only x free as appropriate. Give modal tableau proofs of each one.

2. $\neg\Diamond\neg\varphi \rightarrow \Box\varphi$.