2

MODAL PROOF THEORY

Melvin Fitting

1 INTRODUCTION

We have an interest in those modal formulas that are valid, relative to some suitable notion of validity. But verifying directly that a formula meets a validity condition is generally non-constructive. In part to get around this non-constructivity, formal proof procedures have been created, using a rich variety of mechanisms. A formal proof is a finitary certificate of validity for a formula, and a proof procedure is a specification of the requirements for being a proof. A proof procedure is sound if only valid formulas have proofs—we probably would say an unsound proof procedure is simply not a proof procedure. A proof procedure is complete if all valid formulas have proofs. For modal logics, historically, proof procedures preceded semantics, so the description above is a little anachronistic. But this is not an historical account, and anyway relational semantics is now well-developed, so let us continue as if history never happened.

It will be helpful to settle some terminology first. We assume we have an infinite list of *propositional letters*, typically P, Q, ... Formulas are built up from these in the usual way. For the time being we take as primitive implication ($\supset$), falsehood ($\bot$), and necessity ($\Box$), with negation defined by $\neg X = (X \supset \bot)$, truth by $\top = \neg\bot$, disjunction by $(X \vee Y) = (\neg X \supset Y)$, conjunction by $(X \wedge Y) = \neg(X \supset \neg Y)$, equivalence by $(X \equiv Y) = ((X \supset Y) \wedge (Y \supset X))$, and possibility by $\Diamond X = \neg\Box\neg X$. We'll use X, Y, ... for arbitrary formulas.

A *normal modal logic* is a set of formulas $\mathbf{L}$ meeting the following conditions. First, $\mathbf{L}$ contains all tautologies and all instances of the formula $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$. Second, $\mathbf{L}$ contains Y if it contains X and $X \supset Y$. Third, $\mathbf{L}$ contains $\Box X$ if it contains X. Fourth and finally, with each formula X, $\mathbf{L}$ also contains all substitution instances of X—the result of uniformly replacing propositional letters with more complex modal formulas.

A large variety of formal proof procedures have been created over the years. No proof procedure suffices for every normal modal logic. Well then, what about semantically determined ones? Given any collection of frames, it is not hard to see that the set of formulas valid in all of them is a normal modal logic. No proof procedure suffices for every normal modal logic determined by a class of frames. Certain families of frames meeting special mathematical conditions determine normal logics that have had applications, and these have been given standard names—the same names are commonly used for the frame families and for the normal modal logics they determine. These normal logics tend to have proof procedures, though not every kind of proof procedure may be applicable, even to the most used of these logics. Table 1 shows the frame conditions that are most common in the literature. When traditional names are available I have employed them, but other naming conventions are in use as well. For instance, **B** is also known as **KTB**. In this chapter I will present several kinds of proof procedures, using the logics of Table 1 as examples. I will not attempt to say, for each proof procedure, exactly what range of logics it is good for. Such things are often difficult to determine. But some proof procedures apply to a fairly broad range of normal logics, others to a narrower range. Some provide proofs that humans find intuitively appealing, others are better for machine implementation. I merely wish to display something of the variety available.

Name	Frame Condition
K	none
T	reflexive
K4	transitive
S4	reflexive, transitive
KB	symmetric
B	reflexive, symmetric
S5	reflexive, transitive, symmetric
D	serial
KD4	serial, transitive

Table 1. Some Frame Families for Normal Modal Logics

2 MODAL AXIOMATICS

Axiomatic proof procedures are perhaps the easiest to explain to people. Rules are simple to state and motivate. Candidates for proofs are easily checked for correctness. Unfortunately, axiomatic proofs are generally hard to discover. Today, when automatibility of proof procedures is an important concern, axiomatic systems receive increasingly short shrift. Nonetheless, axiomatic characterizations often make it relatively easy to compare modal logics, and knowing the axioms and rules for a logic supplies a special understanding, even if one does not spend much time constructing axiomatic proofs. And there are modal logics with axiom systems but no decent automatable proof procedures. Let us begin our discussion of proof procedures with axiom systems, then.

An axiomatic proof is a sequence of formulas, each of which is from a specified collection, called *axioms*, or follows from earlier terms of the sequence by a *rule of derivation*. An axiomatic proof proves its last line, or equivalently, proves each of its lines. A proved formula is a *theorem* of the axiomatic system. Of course there is an *effectiveness* requirement—we should be able to tell whether a formula is an axiom or not, and whether a rule of inference is applicable or not. This will be obvious for the axiom systems considered here. Axiom systems differ from each other in the choice of axioms and rules of derivation. They also differ in which propositional connectives and modal operators are taken as primitive, but this is not a deep issue. Early modal axiom systems differed considerably from modern ones in their choices, but this is not an historical account. All current axiom systems for normal modal logics follow the style introduced in [31], so this will be the approach here.

Axioms are particular formulas. It is common to specify them by giving *axiom schemes*. An axiom scheme is a pattern, and any formula matching that pattern is an axiom. When axiom schemes are used, typically a proof procedure will have a finite number of axiom schemes but an infinite number of axioms. An alternative method is to specify a finite number of axioms, and adopt substitution of formulas for propositional letters as an explicit rule of inference. This tends to be more complicated, and we will follow the axiom scheme approach.