# Simple Tableaus for Simple Logics

Melvin Fitting

The Graduate Center, City University of New York,
Departments of Philosophy, Computer Science, Mathematics (emeritus)
melvin.fitting@gmail.com

May 21, 2024

**Abstract**

Consider those many-valued logic models in which the truth values are a lattice that supplies interpretations for the logical connectives of conjunction and disjunction, and which has a De Morgan involution supplying an interpretation for negation. Assume the set of designated truth values is a prime filter in the lattice. Each of these structures determines a simple many-valued logic. We show there is a single Smullyan style signed tableau system appropriate for all of the logics these structures determine. Differences between the logics are confined entirely to tableau branch closure rules. Completeness, soundness, and interpolation can be proved in a uniform way for all cases. Since branch closure rules have a limited number of variations, in fact all the semantic structures determine just four different logics, all well-known ones. Asymmetric logics such as strict/tolerant, $\mathsf{ST}$, also share all the same tableau rules, but differ in what constitutes an initial tableau. It is also possible to capture the notion of *anti-validity* using the same set of tableau rules. Thus a simple set of tableau rules serves as a unifying and classifying device for a natural and simple family of many-valued logics.

## 1 Introduction

The standard presentation for a many-valued logic involves a specification of what the truth values are, of how the logical connectives are to be interpreted on them, and of a subset of truth values called *designated*. A valuation $v$ in such a structure is a mapping from propositional letters to truth values, and extends uniquely to all formulas using the specified interpretation of the connectives. The extension of $v$ to all formulas is commonly called $v$ too, and we tacitly assume this throughout. A set $\Gamma$ of formulas as premises has a set $\Delta$ of formulas as consequences if every valuation $v$ mapping all members of $\Gamma$ to designated values maps some member of $\Delta$ to a designated value. In this way we have a Scott style consequence relation, which we write as $\Gamma \longrightarrow \Delta$. As noted, all this is standard material.

In this paper we work with a simple family of truth value structures of the sort just described. We take truth values to be members of a De Morgan lattice (actually, we do not need distributivity, and will not be assuming it). We take the interpretations of the connectives to be given by the lattice operations, and the De Morgan involution. And finally, we assume that a designated set has the structure of a prime filter. For convenience all these terms will be properly defined later on, though they are standard items.

We create a Smullyan style tableau system that is appropriate for the structures just described. Tableau rules are the same throughout. The only differences from one lattice/filter structure to another comes in the tableau closure rules. Soundness and completeness can be proved uniformly, once and for all. Similarly for interpolation theorems. But it turns out that ultimately the results are parsimonious rather than generous. Since the only tableau differences from structure to structure come in the closure rules, and since there are only a limited number of possibilities for these rules, the logics we can get are exactly: $\mathsf{CL}$ (classical logic), $\mathsf{K}_3$ (Kleene's strong three valued logic), $\mathsf{LP}$ (Priest's logic of paradox), and $\mathsf{FDE}$ (first degree entailment). To some extent this accounts for the ubiquity of these logics in the literature. But we now have simple uniform tableau systems for all four of these logics.

We go on to show that there are simple modifications to what we just briefly described that provide tableau systems for the logics $\mathsf{ST}$ (strict/tolerant logic), $\mathsf{TS}$ (tolerant/strict logic), and $\mathsf{S3}$ (symmetric three valued). The branch extension rules remain the same as before, but the starting and closing rules are where differences appear. We also consider tableau setups for establishing *antivalidity*, an important notion in this area.

## 2 The General Setting

To repeat a bit of what was said in the Introduction, many valued logics are commonly specified by giving some space of truth values, some interpretation of the connectives in that space, and some subset of it whose values are called *designated*. We narrow this generality considerably, to a very well behaved family of structures. And for this we use familiar lattice machinery. All our spaces of truth values will be De Morgan lattices, where distributivity conditions may or may not hold. We will be a bit informal. Strictly speaking, an algebra is specified by saying what its operations are, and a lattice is specified by conditions on its ordering. Thus there are De Morgan algebras and De Morgan lattices. Since each fully determines the other, we are quite casual about the *lattice/algebra* distinction.

### 2.1 Logical Morgan Lattices

The notion of a *logical* Morgan lattice, discussed in this section, comes from earlier work of mine, [13], but the general form derives from the logical bilattices of [2].

**Definition 2.1 (De Morgan and Morgan Lattices)** *A* De Morgan lattice *is a bounded distributive lattice with a De Morgan involution. We call a structure that is like a De Morgan lattice but without requiring distributivity a* Morgan lattice*. Trivially every De Morgan lattice is a Morgan lattice.*

We generally write $\leq$ for a Morgan lattice ordering, $f$ and $t$ for the lower and upper bounds of the lattice, and $\neg$ for the De Morgan involution. Being a lattice, binary meets and joins exist, and we write $\wedge$ and $\vee$ for these. We are overloading notation since we will also use these symbols syntactically as part of our formal logical language, but no confusion is likely to result. A De Morgan involution reverses the ordering, and thus meets the conditions that $a \leq b$ if and only if $\neg b \leq \neg a$, and $\neg\neg a = a$. It follows, using greatest lower and least upper bound properties, that $\neg(a \wedge b) = \neg a \vee \neg b$ and $\neg(a \vee b) = \neg a \wedge \neg b$. "Morgan lattice" is not standard terminology—indeed there does not seem to be a standard terminology here. Dropping the *De* is to suggest dropping *d*istributivity.

Morgan lattices provide us with natural many valued truth value spaces, and accompanying interpretations for logical connectives $\wedge$, $\vee$, and $\neg$ as meet, join, and Morgan involution, and we assume this is how formulas are evaluated from now on. We will also have implication, $\supset$, but it will be defined from other connectives in the familiar way.

We will be working with subsets consisting of designated truth values, and with subsets that are not these but that have direct relationships with them. Of course we want all of them to have some natural structural properties. The following is absolutely standard.

**Definition 2.2 (Filter)** *A non-empty subset $F$ of a lattice is a filter if:*

1. *$F$ is upward closed: $x \in F$ and $x \leq y$ imply $y \in F$;*
2. *$F$ is closed under finite meets; $x, y \in F$ imply $x \wedge y \in F$.*

*$F$ is a prime filter if also:*

3. *$x \vee y \in F$ implies $x \in F$ or $y \in F$.*

In addition to filters, the standard dual notion will also be useful.

**Definition 2.3 (Ideal)** *A non-empty subset $I$ of a lattice is an ideal if:*

1. *$I$ is downward closed: $y \in I$ and $x \leq y$ imply $x \in I$,*
2. *$I$ is closed under finite joins; $x, y \in I$ imply $x \vee y \in I$.*

*$I$ is a prime ideal if also:*

3. *$x \wedge y \in I$ implies $x \in I$ or $y \in I$.*

Filters and ideals are called *proper* if they are not empty and not the entire space. All filters and ideals here will be proper, and we will not say this each time. Also the following alternative characterizations are useful.

**Proposition 2.4** *Let $S$ be an arbitrary subset of a lattice.*

1. *$S$ is a filter if and only if: $x \in S$ and $y \in S \Longleftrightarrow x \wedge y \in S$.*
2. *$S$ is a prime filter if and only if both:*
   (a) *$x \in S$ and $y \in S \Longleftrightarrow x \wedge y \in S$,*
   (b) *$x \in S$ or $y \in S \Longleftrightarrow x \vee y \in S$.*
3. *$S$ is an ideal if and only if: $x \in S$ and $y \in S \Longleftrightarrow x \vee y \in S$.*
4. *$S$ is a prime ideal if and only if both:*
   (a) *$x \in S$ and $y \in S \Longleftrightarrow x \vee y \in S$,*
   (b) *$x \in S$ or $y \in S \Longleftrightarrow x \wedge y \in S$.*

We need some simple operations, mapping subsets of a Morgan lattice to other subsets.

**Definition 2.5** *Let $\mathsf{M} = \langle M, \leq, \neg \rangle$ be a Morgan lattice where $\leq$ is the lattice ordering and $\neg$ is the involution, and let $S \subseteq M$. Then $\overline{S} = \{x \in M \mid x \notin S\}$, and $\neg S = \{\neg x \in M \mid x \in S\}$.*

A simple but useful fact: $\neg y \in S$ if and only if $y \in \neg S$. Here is the verification. First, by definition, $x \in S$ if and only if $\neg x \in \neg S$. Taking $x$ to be $\neg y$, $\neg y \in S$ if and only if $\neg\neg y \in \neg S$. Since $\neg$ is an involution, $\neg\neg y = y$. We thus have $\neg y \in S$ if and only if $y \in \neg S$.

There are some well-known connections between filters and ideals in lattices. There are also a few more that are perhaps less well-known, for Morgan lattices.

**Proposition 2.6** *Met* $\mathsf{M} = \langle M, \leq, \neg\rangle$ *be a Morgan lattice, and let* $S \subseteq M$. *Then we have the following.*

1. *$S$ is a filter if and only if $\overline{S}$ is an ideal.*
2. *$S$ is a prime filter if and only if $\overline{S}$ is a prime ideal.*
3. *$S$ is a filter if and only if $\neg S$ is an ideal.*
4. *$S$ is a prime filter if and only if $\neg S$ is a prime ideal.*
5. *$\overline{\neg S} = \neg\overline{S}$.*

**Proof** Items 1 and 2 are standard for lattices in general. The proofs make use of the actual De Morgan laws for sets. For 3 and 4, the arguments are similar but with the Morgan conditions of the lattice taking over from the actual De Morgan laws. We leave these to you.

Finally, for item 5.

$$\begin{aligned} x \in \overline{\neg S} &\Longleftrightarrow x \notin \neg S \\ &\Longleftrightarrow \neg x \notin S \\ &\Longleftrightarrow \neg x \in \overline{S} \\ &\Longleftrightarrow x \in \neg\overline{S} \end{aligned}$$

■

And now, our central construct. It is at this level that we will introduce tableau systems, and prove completeness, soundness, and interpolation. Think of $M$ as a space of truth values and of $D$ as the designated ones.

**Definition 2.7 (Logical Morgan Lattice)** *Let* $\mathsf{M} = \langle M, \leq, \neg\rangle$ *be a Morgan lattice and $D$ be a subset of $M$ that is a filter. We call the pair* $\langle \mathsf{M}, D\rangle$ *a* logical Morgan lattice. *It is a* prime *logical Morgan lattice if, in addition, $D$ is a prime filter.*

## 2.2 Formal Language and Valuations

We use a standard propositional language. We have a set of *propositional letters*, also called propositional atoms, or just atoms. Formulas are then built up in the usual way with connectives $\wedge$, $\vee$, $\neg$, and $\supset$. This notation plays a familiar double role, with the first three connectives being both formal symbols of the language and also representing Morgan lattice meet, join, and involution. Context will sort things out. We generally use $X$, $Y$, $Z$ with or without subscripts when we represent arbitrary formulas. We use $A$, $B$, $P$, and the like when atomic formulas are meant. This is simply a convenient informal convention and has no deep significance.

**Definition 2.8 (Valuation)** *A* valuation *in a Morgan lattice* $\mathsf{M} = \langle M, \leq, \neg\rangle$ *is a mapping from propositional letters to* $M$*. Every valuation in* $\mathsf{M}$ *extends uniquely to a mapping from* all *formulas to* $M$ *using the following conditions, where on the left appears logical syntax and on the right the Morgan operations.*

$$\begin{aligned} v(\neg X) &= \neg v(X) \\ v(X \wedge Y) &= v(X) \wedge v(Y) \\ v(X \vee Y) &= v(X) \vee v(Y) \\ v(X \supset Y) &= \neg v(X) \vee v(Y) \end{aligned}$$

Note that the first three above are direct; the last essentially takes implication as a defined connective. Since we are working in a Morgan lattice, it would be equivalent if we were to use $\neg(v(X) \wedge \neg v(Y))$.

Next we turn to *validation*, and here some things have changed in recent years. For one thing, instead of just validity for single formulas being the main interest, one now commonly sees *consequence relations*, and the notion of a *consequent* is taken as basic. We understand consequence in the Scott sense. That is, for sets of formulas, $\Gamma$ and $\Delta$, a consequent $\Gamma \longrightarrow \Delta$ is informally understood to say that if we somehow have all the members of $\Gamma$, then we must have at least one of the members of $\Delta$. Derivatively, validity for a formula $X$ is defined as validity of the consequent $\longrightarrow X$.

The second change, another recent one, is to sometimes allow premises (left of a consequent arrow) and conclusions (right of a consequent arrow) to be judged by different standards. We will see the logics $\mathsf{ST}$ and $\mathsf{TS}$ later on; these are logics of this asymmetric sort, and are growing in familiarity. There seems to be no standard name for such logics; we call them *asymmetric* logics and, by contrast we will call standard logics, such as $\mathsf{K}_3$ or $\mathsf{LP}$ *symmetric* logics.

When working with many valued logics it is common to set aside a subset of truth values and call them *designated*. We will do that for each prime logical Morgan lattice $\langle M, D\rangle$, specifically we will take $D$ to be the designated set, giving us a many-valued logic semantics of the symmetric sort. Later on we will also set aside a second set, $\overline{\neg D}$ (notation from Definition 2.5), and consider the asymmetric logic where $D$ is the notion of designation to be used left of the sequent arrow and $\overline{\neg D}$ is the notion of designation to be used right of the arrow. (It will become clear where this somewhat odd definition comes from.) By definition, $D$ is a prime filter, and it follows from Proposition 2.6 that $\overline{\neg D}$ is also a prime filter.

**Definition 2.9 (Validation)** *Let* $\langle \mathsf{M}, D\rangle$ *be a prime logical Morgan lattice. A valuation* $v$ validates *the sequent (or consequent)* $\Gamma \longrightarrow \Delta$ *in* $\langle \mathsf{M}, D\rangle$ *provided:* $v(X) \in D$ *for every* $X \in \Gamma$ *implies* $v(Y) \in D$ *for some* $Y \in \Delta$*. We symbolize this by* $\langle \mathsf{M}, D\rangle \models_v \Gamma \longrightarrow \Delta$*. A sequent is simply* valid *in* $\langle \mathsf{M}, D\rangle$ *if* every *valuation validates it in* $\langle \mathsf{M}, D\rangle$*, and we will write this as* $\langle \mathsf{M}, D\rangle \models \Gamma \longrightarrow \Delta$*.*

*A valuation* $v$ asymmetrically validates $\Gamma \longrightarrow \Delta$ *in* $\langle \mathsf{M}, D\rangle$ *provided:* $v(X) \in D$ *for every* $X \in \Gamma$ *implies* $v(Y) \in \overline{\neg D}$ *for some* $Y \in \Delta$*. We symbolize this by* $\langle \mathsf{M}, D, \overline{\neg D}\rangle \models_v \Gamma \longrightarrow \Delta$*. A sequent is* asymmetrically valid *in* $\langle \mathsf{M}, D\rangle$ *if* every *valuation asymmetrically validates it in* $\langle \mathsf{M}, D\rangle$*, and we will write this as* $\langle \mathsf{M}, D, \overline{\neg D}\rangle \models \Gamma \longrightarrow \Delta$*.*

We conclude the section with some remarks on the connective $\supset$. We have allowed it, and defined its semantic behavior so that $P \supset Q$ is like $\neg P \vee Q$, the familiar material conditional. This is not always a good choice for an implication connective. For instance the sequent $P, P \supset Q \longrightarrow Q$, formalizing *modus ponens*, is not valid in $\mathsf{LP}$, though it is in $\mathsf{K}_3$, while $P \supset Q, Q \supset R \longrightarrow P \supset R$, formalizing *transitivity* or *cut*, is not valid in $\mathsf{K}_3$, though it is in $\mathsf{LP}$. Neither is valid in $\mathsf{FDE}$. A good choice of an implication connective for these logics is a complicated issue that we do not address here.

## 2.3 Signed Formulas and Logical Morgan Lattices

We have the algebraic background and a general semantics in place. We now start on tableau proof systems in a general setting, moving to specific examples in Section 3. A tableau proof system is, essentially, a refutation system in which a proof is a tree constructed along the following lines. To show that a formula is a validity, one begins by supposing it is not. One starts the construction of a tree by placing at the root something that embodies that non-validity supposition. One then 'grows' the tree using *branch extension rules.* Each branch is understood conjunctively, and the tree itself as the disjunction of its branches. A branch is called *closed* if in some sense it represents an impossible situation. If all branches are closed, the tree itself is closed. A closed tree establishes that the initial assumption cannot be the case. Since the assumption was that some formula is not valid, the conclusion is that validity of the formula has been established. This is a very general and very loose description, but formal details will appear shortly.

In the well-known Smullyan book, [28], two versions of tableaus for classical logic were presented, *unsigned* and *signed.* With unsigned tableaus it is simply formulas that appear at tableau nodes, and a proof of $X$ is a closed tableau beginning with $\neg X$. This is fine if one is dealing with classical logic or a classically based logic. It is what is used in [19] for instance, where the logics are modal. But Smullyan also introduced a second version using *signed* formulas, $T\,X$ and $F\,X$. This signed version can be generalized in ways that the unsigned one cannot.

Smullyan showed the equivalence of tableau and sequent presentations. Indeed, he showed that from a suitably abstract point of view they are the same thing. But this equivalence is more complicated if unsigned formulas are used in tableaus. His idea was that a Gentzen sequent, say $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$, corresponds to a set of signed formulas, in this case $\{T\,X_1, \ldots, T\,X_n, F\,Y_1, \ldots, F\,Y_k\}$, in a very natural way. Loosely, $T$ corresponds to being left of arrow and $F$ corresponds to being right of arrow. But without signs available, should $\{X, \neg Y\}$ correspond to $X \longrightarrow Y$ or to $X, \neg Y \longrightarrow$? Or for that matter, what about $\neg Y \longrightarrow \neg X$ The ambiguity vanishes if a distinction is made between $\neg X$ and $F\,X$.

Smullyan only treated classical logic, but as far back as [16] a Smullyan style tableau system for intuitionistic logic was given. In it there was a critical distinction made between $\neg X$, involving intuitionistic negation, and $F\,X$, informally representing that $X$ has not been proved. This meant that, unlike with classical logic, the use of signs could not be avoided. We now have another non-classical reason for using signs. We want to vary the interpretation of them from time to time, while keeping the formulas to which they are applied unchanged. Consequently only signed formula versions of tableaus will appear in what follows.

In addition to the connections with sequents, Smullyan's use of $T$ and $F$ as tableau signs can be seen to have a direct semantic role. Classically, where the set of truth values is just $\{t, f\}$, the signed formula $T\,X$ informally represents that $X$ has the value $t$, and $F\,X$ represents that $X$ has the value $f$ (with respect to some valuation that remains in the background). It is hard to see in the very simple classical setting, but each sign actually has two quite different semantic roles to play. First, each of $T$ and $F$ represents the (classical) negation of the other. Second, in the space of classical truth values each sign represents the complement of what the other sign represents. When the space of truth values is bigger than just two, these roles split apart, and two signs are no longer sufficient. Following [7], we will use four signs: $T$, $F$, $\overline{T}$, and $\overline{F}$. Loosely, $F$ and $T$ will be related through negation, and $F$ and $\overline{F}$ will be related through complementation, as will $T$ and $\overline{T}$. To put it simply, we have machinery to distinguish syntactically between *true* ($T$) and *not false* ($\overline{F}$), and likewise between *false* ($F$) and *not true* ($\overline{T}$).

**Definition 2.10 (Signed Formulas)** *If $X$ is a formula, then $T\,X, F\,X, \overline{T}\,X$, and $\overline{F}\,X$ are signed formulas.*

Above we read the sign $T$ classically as a representation of having truth value $t$, and analogously for $F$. In the more general setting of an arbitrary Morgan lattice, we will take $T$ as representing the set of designated truth values, and $F$ as representing the set of their negations. Then $\overline{T}$ and $\overline{F}$ will represent the compliments of what $T$ and $F$ represent. The following gives this official status. We remind you of Definition 2.5.

**Definition 2.11 (Sign Interpretation)** *Let $D$ be a filter in the Morgan lattice $\mathsf{M} = \langle M, \leq, \neg \rangle$. The truth value sets that each sign represents are given by the following.*

$$\begin{aligned} [\![T]\!] &= D \\ [\![F]\!] &= \neg D \\ [\![\overline{T}]\!] &= \overline{D} \\ [\![\overline{F}]\!] &= \overline{\neg D} \end{aligned}$$

Making use of Proposition 2.6, since $D$ is a filter (prime filter), then $[\![T]\!]$ and $[\![\overline{F}]\!]$ are filters (prime filters) and $[\![F]\!]$ and $[\![\overline{T}]\!]$ are ideals (prime ideals).

**Definition 2.12 (Signed Formula Validation)** *Let $\langle \mathsf{M}, D \rangle$ be a logical Morgan lattice, and let $S\,X$ be a signed formula, where $S$ is one of $T$, $F$, $\overline{T}$, or $\overline{F}$. We say a valuation $v$ in $\mathsf{M}$* validates *the signed formula $S\,X$ if $v(X) \in [\![S]\!]$.*

Note that validation for signed formulas does not depend on the designated set (or on the designated sets in the asymmetric case.)

## 2.4 Tableaus In Logical Morgan Lattices

To characterize a tableau system there are three things we must do: say how a tableau starts, say how a tableau grows, and say how a tableau finishes. We begin with how a tableau starts, then move directly to the finish, and finally discuss the branch extension rules. This presentation of the branch extension rules is the longest part.

### How Tableaus Start

We begin with symmetric many-valued logics, that is, we have a single designated set $D$ of truth values used for both sides of a sequent. As we noted above, tableaus actually work backwards. To prove a sequent $\Gamma \longrightarrow \Delta$, we begin with something that, intuitively at least, represents the possibility that there might be a valuation failing to validate the sequent. In other words, all members of $\Gamma$ could have truth values in $D$, while all members of $\Delta$ could have truth values outside $D$, that is, in $\overline{D}$. Using the sign interpretations from Definition 2.11, being in $D$ corresponds to a sign of $T$, while being in $\overline{D}$ corresponds to a sign of $\overline{T}$. This gives us the following.

**Definition 2.13 (Initial Tableau for Sequents In Symmetric Logics)** *An initial tableau for an attempted proof of $\Gamma \longrightarrow \Delta$ in $\langle \mathsf{M}, D \rangle$ contains signed formulas $T\,X_1, \ldots, T\,X_n$, $\overline{T}\,Y_1$, $\ldots$, $\overline{T}\,Y_k$, where each $X_i \in \Gamma$ and each $Y_j \in \Delta$.*

Next we turn to asymmetric systems. Here a failure of $\Gamma \longrightarrow \Delta$ to be valid means that members of $\Gamma$ might have truth values in the left designated set $D$, while members of $\Delta$ fail to have members in the right designated set, $\overline{\neg D}$, and thus have truth values in its complement, $\neg D$. As before, $D$ corresponds to a sign of $T$, but now we have $\neg D$, which corresponds to a sign of $F$.

**Definition 2.14 (Initial Tableau for Sequents In Asymmetric Logics)** *An initial tableau for an attempted proof of* $\Gamma \longrightarrow \Delta$ *in* $\langle \mathsf{M}, D, \overline{\neg D}\rangle$ *contains signed formulas* $T\,X_1, \ldots, T\,X_n, F\,Y_1, \ldots,$ $F\,Y_k$*, where each* $X_i \in \Gamma$ *and each* $Y_j \in \Delta$*.*

## How Tableaus End

Closure of a tableau is meant to represent a contradiction, an impossible situation. This depends to some extent on the details of the algebraic structure of truth values, but the general idea is, an impossible situation on a branch is one in which the same formula appears with two signs that cannot be mutually held. Thus, for instance, $T$ and $\overline{T}$ cannot both apply to the same formula because by Definition 2.11, $T\,X$ represents a situation where the truth value of $X$ is in the designated set $D$, while $\overline{T}\,X$ says $X$ has a value outside of $D$. Here are the formalities.

A tableau might end without closing because we have run out of new things to do. We will see that such tableau constructions provide us with counter-models. Though it is possible for tableaus to run forever in some logics, it can be shown that this is not the case with the logics we look at here.

**Definition 2.15 (Tableau Closure)** *The following is for a tableau in* $\langle \mathsf{M}, D\rangle$*. A tableau branch is* closed *if, for some formula* $X$*, the branch contains two signed formulas,* $S_1\,X$ *and* $S_2\,X$*, where* $[\![S_1]\!] \cap [\![S_2]\!] = \emptyset$*. The branch is* atomically *closed if* $X$ *is atomic. A tableau is (atomically) closed if every branch is (atomically) closed. A* tableau proof *of a sequent in* $\langle \mathsf{M}, D\rangle$ *is a closed tableau in* $\langle \mathsf{M}, D\rangle$ *that begins according to Definitions 2.13 and 2.14.*

It is always the case that a branch is closed if it contains a formula having both the signs $T$ and $\overline{T}$, or both $F$ and $\overline{F}$. These may not be the only cases; it depends on the details of the designated set, and thus of the logical Morgan lattice.

Every closed tableau can be continued to one that is *atomically closed.* In fact it is rather easy to prove that a branch that is closed using a formula of degree $n > 0$ can be extended to one or more closed branches for which the degrees of formulas involved in closure are of lower degree than $n$. The reader might give this a try once the branch extension rules have been given. Equivalently one can handle this issue non-constructively. In fact we will show that we have tableau soundness allowing non-atomic closure, and tableau completeness requiring atomic closure. It follows that if a formula $X$ has a proof allowing non-atomic closure, it is valid, and hence has a proof in which closure is atomic.

## How Tableaus Grow

Tableau rules are the same for all logics based on prime logical Morgan lattices, and this is also independent of whether we have one designated set or two. In [28] *uniform notation* was used as a way of grouping similar cases of signed formulas together. This becomes even more helpful when four signs are available, not just two. Smullyan grouped the usual binary connectives into the *conjunctives*, his $\alpha$ case, and *disjunctives*, his $\beta$ case. (There were also $\gamma$ and $\delta$ cases for universal and existential quantifiers, but they aren't needed here.) Smullyan chose to include signed formulas whose main connective was negation in both the $\alpha$ and $\beta$ groups, but we prefer to omit it from both. Instead we introduce a new unary case for them, $\rho$, which is meant to suggest "reverse". This is not a deep difference.

**Definition 2.16 (Uniform Notation)** *We have four signs, $T$, $F$, $\overline{T}$, and $\overline{F}$, and four corresponding signed formula types, $T\,X$, $F\,X$, $\overline{T}\,X$, and $\overline{F}\,X$, where $X$ is a formula. Each unary signed formula is in the $\rho$ category, and each signed formula involving a binary connective is in the $\alpha$, or conjunctive, category or it is in the $\beta$, or disjunctive, category. For each $\rho$ signed formula a* component, *$\rho_0$ is defined, and for each binary case two* components *are defined: $\alpha_1$ and $\alpha_2$ for the $\alpha$ case, and $\beta_1$ and $\beta_2$ for the $\beta$ case. All this is specified in the following table.*

| $\rho$ | $\rho_0$ |
|---|---|
| $T\,\neg X$ | $F\,X$ |
| $F\,\neg X$ | $T\,X$ |
| $\overline{T}\,\neg X$ | $\overline{F}\,X$ |
| $\overline{F}\,\neg X$ | $\overline{T}\,X$ |

| $\alpha$ | $\alpha_1$ | $\alpha_2$ | $\beta$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|---|---|
| $T\,X \wedge Y$ | $T\,X$ | $T\,Y$ | $F\,X \wedge Y$ | $F\,X$ | $F\,Y$ |
| $F\,X \vee Y$ | $F\,X$ | $F\,Y$ | $T\,X \vee Y$ | $T\,X$ | $T\,Y$ |
| $F\,X \supset Y$ | $T\,X$ | $F\,Y$ | $T\,X \supset Y$ | $F\,X$ | $T\,Y$ |
| $\overline{F}\,X \wedge Y$ | $\overline{F}\,X$ | $\overline{F}\,Y$ | $\overline{T}\,X \wedge Y$ | $\overline{T}\,X$ | $\overline{T}\,Y$ |
| $\overline{T}\,X \vee Y$ | $\overline{T}\,X$ | $\overline{T}\,Y$ | $\overline{F}\,X \vee Y$ | $\overline{F}\,X$ | $\overline{F}\,Y$ |
| $\overline{T}\,X \supset Y$ | $\overline{F}\,X$ | $\overline{T}\,Y$ | $\overline{F}\,X \supset Y$ | $\overline{T}\,X$ | $\overline{F}\,Y$ |

Now we show that uniform notation really does identify similarly behaving cases.

**Proposition 2.17** *Let $\langle \mathsf{M}, D\rangle$ be a prime logical Morgan lattice, where $\mathsf{M} = \langle M, \le, \neg\rangle$, and let $v$ be a valuation in $M$.*

$$\begin{aligned} &v \text{ validates } \rho \iff v \text{ validates } \rho_0\\ &v \text{ validates } \alpha \iff v \text{ validates } \alpha_1 \text{ and } v \text{ validates } \alpha_2\\ &v \text{ validates } \beta \iff v \text{ validates } \beta_1 \text{ or } v \text{ validates } \beta_2\end{aligned}$$

**Proof** Uniform notation condenses a large number of cases. We look at some representatives of them; the rest are similar.

**Case** $F\,\neg X$ This is a $\rho$ signed formula.

$$\begin{aligned} v \text{ validates } F\,\neg X &\iff v(\neg X) \in [\![F]\!] \\ &\iff \neg v(X) \in [\![F]\!] && \text{Definition 2.8}\\ &\iff \neg v(X) \in \neg D && \text{Definition 2.11}\\ &\iff v(X) \in D \\ &\iff v(X) \in [\![T]\!] && \text{Definition 2.11}\\ &\iff v \text{ validates } T\,X \end{aligned}$$

**Case** $\overline{F}\,\neg X$ This is also a $\rho$ signed formula.

$$\begin{aligned} v \text{ validates } \overline{F}\,\neg X &\iff v(\neg X) \in [\![\overline{F}]\!] \\ &\iff \neg v(X) \in [\![\overline{F}]\!] && \text{Definition 2.8}\\ &\iff \neg v(X) \in \overline{\neg D} && \text{Definition 2.11}\\ &\iff \neg v(X) \in \neg \overline{D} && \text{Proposition 2.6}\\ &\iff v(X) \in \overline{D} \\ &\iff v(X) \in [\![\overline{T}]\!] && \text{Definition 2.11}\\ &\iff v \text{ validates } \overline{T}\,X \end{aligned}$$

**Case** $T\,X \wedge Y$ This is an $\alpha$ signed formula, where $\alpha_1 = T\,X$ and $\alpha_2 = T\,Y$.

$$\begin{aligned} v \text{ validates } T\,X \wedge Y &\iff v(X \wedge Y) \in [\![T]\!] \\ &\iff v(X) \wedge v(Y) \in [\![T]\!] && \text{Definition 2.8} \\ &\iff v(X) \in [\![T]\!] \text{ and } v(Y) \in [\![T]\!] && [\![T]\!] \text{ is a filter} \\ &\iff v \text{ validates } T\,X \text{ and } v \text{ validates } T\,Y \end{aligned}$$

**Case** $F\,X \wedge Y$ This is a $\beta$ signed formula, where $\beta_1 = F\,X$ and $\beta_2 = F\,Y$:

$$\begin{aligned} v \text{ validates } F\,X \wedge Y &\iff v(X \wedge Y) \in [\![F]\!] \\ &\iff v(X) \wedge v(Y) \in [\![F]\!] && \text{Definition 2.8} \\ &\iff v(X) \in [\![F]\!] \text{ or } v(Y) \in [\![F]\!] && [\![F]\!] \text{ is an ideal} \\ &\iff v \text{ validates } F\,X \text{ or } v \text{ validates } F\,Y \end{aligned}$$

**Case** $\overline{T}\,X \vee Y$ This is an $\alpha$ signed formula, where $\alpha_1 = \overline{T}\,X$ and $\alpha_2 = \overline{T}\,Y$:

$$\begin{aligned} v \text{ validates } \overline{T}\,X \vee Y &\iff v(X \vee Y) \in [\![\overline{T}]\!] \\ &\iff v(X) \vee v(Y) \in [\![\overline{T}]\!] && \text{Definition 2.8} \\ &\iff v(X) \in [\![\overline{T}]\!] \text{ and } v(Y) \in [\![\overline{T}]\!] && [\![\overline{T}]\!] \text{ is a prime ideal} \\ &\iff v \text{ validates } \overline{T}\,X \text{ and } v \text{ validates } \overline{T}\,Y \end{aligned}$$

**Case** $\overline{F}\,X \vee Y$ This is a $\beta$ signed formula, where $\beta_1 = \overline{F}\,X$ and $\beta_2 = \overline{F}\,Y$:

$$\begin{aligned} v \text{ validates } \overline{F}\,X \vee Y &\iff v(X \vee Y) \in [\![\overline{F}]\!] \\ &\iff v(X) \vee v(Y) \in [\![\overline{F}]\!] && \text{Definition 2.8} \\ &\iff v(X) \in [\![\overline{F}]\!] \text{ or } v(Y) \in [\![\overline{F}]\!] && [\![\overline{F}]\!] \text{ is a prime filter} \\ &\iff v \text{ validates } \overline{F}\,X \text{ or } v \text{ validates } \overline{F}\,Y \end{aligned}$$

■

As we noted earlier, the Smullyan book [28] gives a now-standard set of tableau rules for classical logic, using $T$ and $F$ as signs, and Smullyan grouped these rules into convenient $\alpha$ and $\beta$ cases. We have extended Smullyan's system with $\overline{T}$ and $\overline{F}$ and a unary case, and we apply the rules in the general context of prime logical Morgan lattices. It turns out that Smullyan's uniform version of tableau rules is still exactly what is needed. What all this gets us is a kind of schematic proof system, relative to a prime logical Morgan lattice semantics. Subsequently we go on to prove soundness and completeness results in this general setting. Now, here are the *branch extension rules*, using the uniform notation of Definition 2.16.

**Definition 2.18 (Tableau Rules, Using Uniform Notation)**

$$\frac{\rho}{\rho_0} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Unwound from the notation we have the following. If a $\rho$ formula occurs on a tableau branch, the branch can be extended by adding $\rho_0$ to the end. If an $\alpha$ is present, the branch can be extended by adding $\alpha_1$ and then $\alpha_2$ to the end. If a $\beta$ is present, the branch end can be forked, with $\beta_1$ added to one fork and $\beta_2$ to the other. Note: as tableau rules, these are *non-deterministic.* They only say what *may* be done, not what *must* be done.

We now state and prove a key fact about the rules: they preserve satisfiability. Indeed, we will prove something stronger: they preserve satisfiability in both directions.

**Definition 2.19 (Strongly Sound)** *We say a tableau rule from Definition 2.18 is* strongly sound, *relative to a given logical Morgan lattice $\langle \mathsf{M}, D\rangle$ provided that for each valuation $v$ in $\langle \mathsf{M}, D\rangle$:*

1. *if the tableau rule is non-branching, $v$ validates the signed formula above the line if and only if $v$ validates the signed formula or formulas below the line;*

2. *if the tableau rule is branching, $v$ validates the signed formula above the line if and only if $v$ validates one of the signed formulas below the line.*

**Proposition 2.20** *Every tableau rule in Definition 2.18 is strongly sound, relative to any prime logical Morgan lattice.*

**Proof** This is what Proposition 2.17 directly gives us. ■

To sum up, for any prime logical Morgan lattice we have specified tableau systems, symmetric and asymmetric. A tableau for a sequent $\Gamma \longrightarrow \Delta$ begins according to Definition 2.13 or Definition 2.14 and it is here that the difference between symmetric and asymmetric logics appears. The tableau grows using the rules in Definition 2.18, which are completely independent of the choice of prime Morgan lattice. If a tableau is produced that is closed according to Definition 2.15, we have given a tableau proof for the sequent. It is only at this point that differences in the lattice structure play a role.

### For Reference: The Full Rule Set

For the convenience of the reader, in Figure 2.1 we give the full and unabbreviated set of tableau rules, expanding the uniform notation used in Definition 2.18. The rules for $T$ and $F$ are exactly the Smullyan rules from [28].

## 2.5 Soundness and Completeness

We now give a uniform proof of tableau soundness and of tableau completeness, relative to any arbitrary prime logical Morgan lattice.

### Soundness

Assume $\langle \mathsf{M}, D\rangle$ is a prime logical Morgan lattice. We will show that a sequent that is not valid in it can not have a tableau proof in the tableau system for it, and this is the case whether we consider things symmetrically or asymmetrically. Then contrapositively, a sequent with a tableau proof must be a valid sequent.

The main tool is the notion of satisfiability, so to begin we must say what this means. A tableau is satisfiable if one of its branches is satisfiable. A branch is satisfiable if there is a valuation that validates every formula on the branch. The heart of the matter is to show that satisfiability is an

$$\frac{T\,\neg X}{F\,X} \qquad \frac{F\,\neg X}{T\,X}$$

$$\frac{\overline{F}\,\neg X}{\overline{T}\,X} \qquad \frac{\overline{T}\,\neg X}{\overline{F}\,X}$$

$$\frac{T\,X \wedge Y}{\begin{array}{c}T\,X\\ T\,Y\end{array}} \qquad \frac{F\,X \vee Y}{\begin{array}{c}F\,X\\ F\,Y\end{array}} \qquad \frac{F\,X \supset Y}{\begin{array}{c}T\,X\\ F\,Y\end{array}}$$

$$\frac{\overline{F}\,X \wedge Y}{\begin{array}{c}\overline{F}\,X\\ \overline{F}\,Y\end{array}} \qquad \frac{\overline{T}\,X \vee Y}{\begin{array}{c}\overline{T}\,X\\ \overline{T}\,Y\end{array}} \qquad \frac{\overline{T}\,X \supset Y}{\begin{array}{c}\overline{F}\,X\\ \overline{T}\,Y\end{array}}$$

$$\frac{T\,X \vee Y}{T\,X \mid T\,Y} \qquad \frac{F\,X \wedge Y}{F\,X \mid F\,Y} \qquad \frac{T\,X \supset Y}{F\,X \mid T\,Y}$$

$$\frac{\overline{F}\,X \vee Y}{\overline{F}\,X \mid \overline{F}\,Y} \qquad \frac{\overline{T}\,X \wedge Y}{\overline{T}\,X \mid \overline{T}\,Y} \qquad \frac{\overline{F}\,X \supset Y}{\overline{T}\,X \mid \overline{F}\,Y}$$

Figure 2.1: Full Tableau Rule Set, Expanded

invariant of tableau construction, that is, satisfiability is preserved by every tableau rule. And, in fact, this is something that Proposition 2.20 specifically tells us is so. Note that this preservation of satisfiability is the case whether we are considering a symmetric or an asymmetric consequence relation using $\langle \mathsf{M}, D\rangle$, since they agree on branch extension rules.

Next we look at how the tableau construction starts, and here things are different if we are working symmetrically or asymmetrically. We consider these one at a time.

First the symmetric case. Let $\Gamma \longrightarrow \Delta$ be a sequent, and assume it is not valid in $\langle \mathsf{M}, D\rangle$. Then there is some valuation $v$ that maps all $X_i \in \Gamma$ to $D = \llbracket T \rrbracket$, but no $Y_j \in \Delta$ to $D$, and thus every $Y_j$ to $\overline{D} = \llbracket \overline{T} \rrbracket$. An attempted tableau proof for this sequent begins with a single branch, containing $T\,X_1, \ldots, T\,X_n, \overline{T}\,Y_1, \ldots, \overline{T}\,Y_k$, where $X_i \in \Gamma$ and $Y_j \in \Delta$. Clearly $v$ satisfies each of these signed formulas, Definition 2.12, so our initial tableau is satisfiable.

Next the asymmetric case. As before, assume $\Gamma \longrightarrow \Delta$ is a sequent that is not valid in $\langle \mathsf{M}, D\rangle$, but now in the asymmetric sense. Then there is a valuation $v$ mapping every member of $\Gamma$ to $D = \llbracket T \rrbracket$, but no member of $\Delta$ to $\overline{\neg D}$, and hence every member of $\Delta$ to $\neg D = \llbracket F \rrbracket$. By Definition 2.14, an attempted tableau proof of the sequent begins with $T\,X_1, \ldots, T\,X_n, F\,Y_1, \ldots, F\,Y_k$, where $X_i \in \Gamma$ and $Y_j \in \Delta$. But $v$ satisfies each of these signed formulas, and so again we are starting with a satisfiable tableau.

Whether we work symmetrically or asymmetrically, an attempted tableau proof for a non-valid sequent begins with a satisfiable initial tableau. Then by the invariance of satisfiability, every subsequent tableau is satisfiable. But it is obvious that a satisfiable tableau cannot be closed. Thus there can be no tableau proof for a non-valid sequent.

## Completeness

There are several ways of proving tableau completeness (requiring that closure always be atomic). We present perhaps the most common version, simply extended to apply to logical Morgan lattices instead of, say, to classical logic only. One constructs a tableau systematically and fairly, eventually applying every applicable rule. Then assuming we have an unprovable sequent, the tableau attempt must fail. From any open branch of the failed tableau, a counter-model to each signed formula on the branch can be constructed, thus establishing completeness (and incidentally, that a systematic construction must succeed if anything does.)

But first, a small but important point. We will establish *strong* completeness, for a sequent involving possibly countably infinite sets of formulas. Of course if a sequent $\Gamma \longrightarrow \Delta$ has countably many formulas, we can't simply start with a tableau containing signed versions of them all. Fortunately a solution taken from [28] is available and is quite simple. Begin stage 1 of the tableau construction by putting down the first signed formula drawn from $\Gamma$ and the first drawn from $\Delta$ (if one is empty, just put a single signed formula down instead of two). Next, carry out one step of whatever your systematic tableau construction procedure might be, then add appropriately signed versions of the second formulas from $\Gamma$ and $\Delta$ to the end of each open branch. Carry out one more step of a systematic procedure, then add signed versions of the third formulas in $\Gamma$ and $\Delta$ to the ends of each open branch, and so on. The general idea should be clear. We interleave adding formulas to branch ends and steps of a systematic construction procedure.

A systematic procedure that eventually does everything possible is really quite easy. Pick the uppermost signed formula on each branch that has had no rule applied to it on that branch, apply the appropriate rule for each branch the occurrence is on, then check the occurrence off on those branches and never use it again. (A signed formula may occur on more than one branch. Each branch is a separate application.) Since every rule application reduces formula degree, we must completely process any given instance of a signed formula on a branch after a finite number of stages of the construction. A more elaborate description of the construction could be given, but this should suffice.

Now with the background out of the way, let us first look at symmetric logics. Suppose we are seeking a tableau proof for $\Gamma \longrightarrow \Delta$. This might involve infinite sets, so we proceed according to the systematic fair approach outlined above. The procedure may terminate with a proof. If not, it may terminate without closure, or it may run forever if infinite sets are involved. Either way, if we don't have closure we wind up with a tableau with at least one open branch, a branch that will be infinite if infinite sets are involved (König's lemma comes in here). We can use such an open branch to construct an appropriate counter-model.

In what follows, $\mathcal{B}$ is an atomically unclosed branch of our tableau (possibly infinite), where the tableau was constructed using a fair algorithm, so everything that could be done has been done.

We note the following simple property of $\mathcal{B}$: for any atomic formula $A$, at most two of $T\,A$, $\overline{T}\,A$, $F\,A$, and $\overline{F}\,A$ can appear on $\mathcal{B}$. Because only one of $T\,A$, $\overline{T}\,A$ can occur, or else the branch would be closed; likewise only one of $F\,A$, $\overline{F}\,A$ can occur, for the same reason.

Now as promised, we use the information on $\mathcal{B}$ to construct a model. We define a valuation $v$ by specifying it on atomic formulas, then extend it to all formulas as usual. Suppose atomic formula $A$ occurs on $\mathcal{B}$ with two different signs. By Definition 2.11, each sign has an associated set of truth values and, since $\mathcal{B}$ is not closed, those sets must overlap. Let $v(A)$ be any member of that overlap. Next, suppose $A$ occurs on $\mathcal{B}$ with only one sign. Then let $v(A)$ be any truth value in the set associated with that sign. Finally, suppose $A$ does not occur on $\mathcal{B}$ with any sign. Then let $v(A)$ be arbitrary. Thus valuation $v$ has been specified, and it extends uniquely to all formulas.

Next one shows that *every* signed formula on branch $\mathcal{B}$ is satisfied by the valuation $v$ just constructed. This is done by induction on degree, and makes use of Proposition 2.20. It is quite straightforward, and we omit the details. Since $T\,X$ is on the branch for every $X$ in $\Gamma$, and $\overline{T}\,Y$ is present for every $Y$ in $\Delta$, these signed formulas are satisfied by $v$, and thus $v$ maps each member of $\Gamma$ to $D$ and each member of $\Delta$ to a value outside $D$. Then $v$ is a counter-model to $\Gamma \longrightarrow \Delta$.

This covers symmetric logics. Asymmetric logics are very similar, and we leave checking the details to the reader.

## 2.6 Anti-Validity and Tableaus

A sequent is valid if every valuation validates it. The dual notion, anti-validity, also plays a significant role, as can be seen in [27] and in [8]. We remind the reader of Definition 2.9 for validation. And, for reasons that will become obvious, we only work with finite sequents.

**Definition 2.21 (Anti-Validity)** *Let $\langle \mathsf{M}, D\rangle$ be a prime logical Morgan lattice. A sequent $\Gamma \longrightarrow \Delta$ is* symmetrically anti-valid *in $\langle \mathsf{M}, D\rangle$ if no valuation validates the sequent. That is, for every valuation $v$, $\langle \mathsf{M}, D\rangle \not\models_v \Gamma \longrightarrow \Delta$. Likewise the sequent is* asymmetrically anti-valid *in $\langle \mathsf{M}, D\rangle$ if no valuation asymmetrically validates the sequent. That is, for every valuation $v$, $\langle \mathsf{M}, D, \overline{\neg D}\rangle \not\models_v \Gamma \longrightarrow \Delta$.*

For instance, in classical logic the sequent $\longrightarrow P \vee \neg P$ is valid, $\longrightarrow P \wedge \neg P$ is anti-valid, and $\longrightarrow P \supset (P \wedge Q)$ is neither valid nor anti-valid. We connect anti-validity with tableaus in this section. Some particular examples that have been important in the literature will be given later.

We start with the symmetric setting. Tableau systems all work backwards in a sense. For instance, speaking informally, to establish validity of a sequent we begin a tableau with signed formulas representing what the situation would be if the sequent were *not* valid. We then show this leads to a contradiction by extending the initial tableau to one that is closed, and we conclude that validity of the sequent has been established.

We now apply this backward idea to anti-validity. Let $\langle \mathsf{M}, D\rangle$ be a prime Morgan lattice. A finite sequent $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$ is symmetrically anti-valid in $\langle \mathsf{M}, D\rangle$ if no valuation in $\langle \mathsf{M}, D\rangle$ validates it. Suppose we hypothesize the sequent is *not* anti-valid. Then some valuation would validate it. That is, there would be some valuation that either maps one of the $X_i$ to a non-designated value (thus inside $\overline{D}$), or maps one of the $Y_j$ to a designated value (thus inside $D$). We want to show each of these possibilities must fail, thus contradicting our hypothesis. Using the sign interpretations from Definition 2.11, we should show closure of $n+k$ tableaus, each starting with one of $\overline{T}\,X_i$, …, $\overline{T}\,X_n$, or $T\,Y_1$, … $T\,Y_k$. Formally stated, this is the following.

**Proposition 2.22 (Symmetric Anti-Validity)** *A sequent $X_1, \ldots, X_n \longrightarrow Y_1 \ldots, Y_k$ is symmetrically anti-valid in $\langle \mathsf{M}, D\rangle$ if and only if there are $n+k$ closed tableaus, one for $\overline{T}\,X_1$, …, one for $\overline{T}\,X_n$, one for $T\,Y_1$, …, and one for $T\,Y_k$.*

Next we look at the asymmetric setting, following the same methodology. Again, let $\langle \mathsf{M}, D\rangle$ be a prime Morgan lattice, but now we work with $\langle \mathsf{M}, D, \overline{\neg D}\rangle$. Following the ideas above, to establish the anti-validity of $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$ we should begin by hypothesizing it is *not* anti-valid in $\langle \mathsf{M}, D, \overline{\neg D}\rangle$, and show this leads to an imposibility. That is, we hypothesize there is some valuation that validates the sequent, and so maps one of the $X_i$ to $\overline{D}$, or maps one of the $Y_j$ to $\overline{\neg D}$. We want each of these to fail. Then using the sign interpretations from Definition 2.11, we must show closure for tableaus starting with each of $\overline{T}\,X_1$, …, $\overline{T}\,X_n$, and $\overline{F}\,Y_1$, … $\overline{F}\,Y_k$.

**Proposition 2.23 (Asymmetric Anti-Validity)** *A sequent $X_1, \dots, X_n \longrightarrow Y_1, \dots, Y_k$ is asymmetrically anti-valid in $\langle \mathsf{M}, D\rangle$ if and only if there are $n+k$ closed tableaus, one for $\overline{T}\, X_1$, ..., one for $\overline{T}\, X_n$, one for $\overline{F}\, Y_1$, ..., and one for $\overline{F}\, Y_k$.*

We have given informal arguments why Propositions 2.22 and 2.23 should hold. These can easily be turned into formal arguments along the lines of the soundness and completeness proofs found in Sections 2.5. We leave this to the reader.

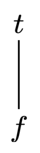
## 3 Logical Morgan Lattice Examples

In this section we look at examples of structures to which the general approach just presented applies. It should be recalled that the tableau branch extension rules are independent of the choice of prime logical Morgan lattice $\langle \mathsf{M}, D\rangle$ and the initial tableau setup only depends on whether we want a symmetric logic or an asymmetric one. It is the closure rules that depend on details of the prime filter $D$. Our examples are: CL, the standard structure for classical logic, a three element lattice used for both of the familiar three-valued logics $\mathsf{K}_3$ and LP, a four-valued logical Morgan lattice that is standard for FDE, first degree entailment. We also look at a less familiar six-valued logical Morgan lattice that happens to be non-distributive, and we look at S3, an intersection logic.

### 3.1 Classical Logic, CL

Classical logic is, of course, the most familiar logic. It is included here to complete the set, so to speak. The discussion also serves to provide a lead-in to our presentation of various topics in less familiar settings. Tableaus for classical logic were thoroughly explored in [28]; what we add to that is simply to check that classical logic, CL, fits into our general treatment.

Classical logic is, famously, two valued. To construct an appropriate prime logical Morgan lattice, we take the set of truth values to be $\{t, f\}$, with the upward ordering shown in Subfigure 3.1a: $f \le t$, $f \le f$, $t \le t$, but not $t \le f$. The Morgan lattice here is $\mathsf{M} = \langle M, \le, \neg\rangle$, where $M$ is the set $\{f, t\}$, $\le$ is the order we just gave, and $\neg$ is the mapping that switches around $t$ and $f$, a De Morgan involution. The prime filter $D$ of designated truth values is $\{t\}$. This structure $\langle \mathsf{M}, D\rangle$ is the simplest example of a prime logical Morgan lattice, in fact, a prime logical *De* Morgan lattice. We call the structure CL, and we also use CL for the logic it determines, namely classical logic.



(a) Two Truth Values

| $\wedge$ | $f$ | $t$ |
|---|---|---|
| $f$ | $f$ | $f$ |
| $t$ | $f$ | $t$ |

| $\vee$ | $f$ | $t$ |
|---|---|---|
| $f$ | $f$ | $t$ |
| $t$ | $t$ | $t$ |

| $\supset$ | $f$ | $t$ |
|---|---|---|
| $f$ | $t$ | $t$ |
| $t$ | $f$ | $t$ |

| $\neg$ | |
|---|---|
| $f$ | $t$ |
| $t$ | $f$ |

(b) Two-Valued Truth Tables

Figure 3.1: Classical Logic, CL

For this classical case, where $D = \{t\}$, we have that $\overline{D} = \neg D = \{f\}$, and $\overline{\neg D} = D$. Using the notation from Definition 2.11, $[\![T]\!] = [\![\overline{F}]\!] = D$ and $[\![F]\!] = [\![\overline{T}]\!] = \overline{D}$. Thus there is redundancy in our four signs, but let us ignore that and proceed formally for the time being. Definition 2.13 tells us how a tableau starts. Closure conditions are supplied by Definition 2.15, and specialize to those in Definition 3.1.

**Definition 3.1 (CL Closure Conditions)** *A* CL *tableau branch is closed if it contains the same formula with any of the following pairs of signs:*

$$\begin{array}{ccc} T & \textit{and} & \overline{T} \\ T & \textit{and} & F \\ \overline{F} & \textit{and} & \overline{T} \\ \overline{F} & \textit{and} & F \end{array}$$

We display a simple example of a closed classical, CL, tableau, in Figure 3.2. It is a proof of the sequent $\neg(X \wedge Y) \longrightarrow (\neg X \vee \neg Y)$.

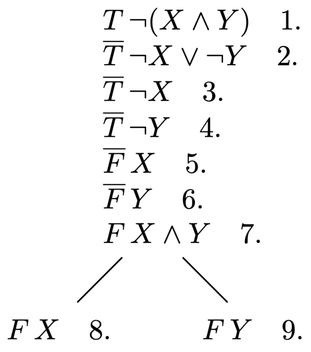


Figure 3.2: Closed Classical Tableau Example

Our general completeness proof in Section 2.5 shows how to construct a countermodel from a failed tableau, provided everything allowed has been done. Figure 3.3 shows how this works in practice. We attempt to construct a tableau proof for $(X \wedge \neg Y) \longrightarrow ((X \supset Y) \wedge (Y \supset X))$, where we assume $X$ and $Y$ are atomic. The right branch is closed because of 3 and 11, or equally because of 5 and 10. But the left branch is not closed. Based on 3 and 8, let $v$ be a valuation such that $v(X) = t$. Similarly $v(Y) = f$ by 5 and 9. It is easy to check that $v$ validates every signed formula on the branch. Then, in particular, $v(X \wedge \neg Y) = t$ and $v((X \supset Y) \wedge (Y \supset X)) = f$, so $v$ does not validate $(X \wedge \neg Y) \longrightarrow ((X \supset Y) \wedge (Y \supset X))$.

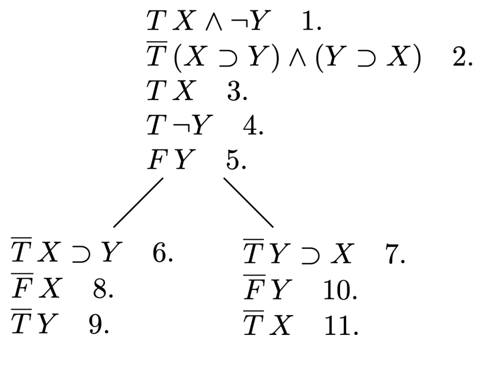


Figure 3.3: Unclosed Classical Tableau Example

We say a few words about *anti-validation*, which was discussed earlier in Section 2.6. There we gave, as a simple example, the sequent $\longrightarrow P \wedge \neg P$, which no valuation can validate. We also gave Proposition 2.22, which related anti-validity to tableaus. Applying that Proposition to our

$$\begin{array}{l} T\, P \wedge \neg P \\ T\, P \\ T\, \neg P \\ F\, P \end{array}$$

Figure 3.4: Anti-Validity Example

example, the sequent $\longrightarrow P \wedge \neg P$ is anti-valid just in case there is a closed tableau beginning with $T\, P \wedge \neg P$. Figure 3.4 shows that simple tableau.

In Smullyan's original signed tableau system for classical logic there were only two signs, $T$ and $F$, and only the rules for these signs in Figure 2.1 were present. Since for $\mathsf{CL}$ we have that $[\![T]\!]$ and $[\![\overline{F}]\!]$ are the same, as are $[\![F]\!]$ and $[\![\overline{T}]\!]$, it looks like Smullyan's version and ours ought to determine the same logic. Of course they do, because Smullyan established the completeness of his system, and completeness of our version follows from the general argument in Section 2.5. But showing equivalence of the two tableau systems this way involves a detour through semantics. One can quite easily be more direct.

**Proposition 3.2** *Let $\mathcal{T}$ be a tableau constructed using the branch extension rules of our four-sign system for* $\mathsf{CL}$*. Let $R(\mathcal{T})$ be the result of replacing in $\mathcal{T}$ all occurrences of $\overline{T}$ with $F$ and all occurrences of $\overline{F}$ with $T$. Then $R(\mathcal{T})$ is also a correctly constructed tableau in* $\mathsf{CL}$*, and $R(\mathcal{T})$ is closed if and only if $\mathcal{T}$ is closed.*

**Proof** A look at the tableau rules in Figure 2.1 shows that the replacement of $\overline{F}$ with $T$ and $\overline{T}$ with $F$ in any tableau rule yields another tableau rule. Similarly in the converse direction. Likewise a look at Definition 3.1 shows that such a replacement turns every $\mathsf{CL}$ closure condition into another $\mathsf{CL}$ closure condition, and again in the converse direction too. ■

Now, suppose we have a proof of a sequent $\Gamma \longrightarrow \Delta$ in $\mathsf{CL}$, so there is a closed $\mathsf{CL}$ tableau $\mathcal{T}$ starting with members of $\Gamma$ with a sign of $T$ and members of $\Delta$ with a sign of $\overline{T}$. Then $R(\mathcal{T})$ will be a closed tableau using only the signs $T$ and $F$, but starting with members of $\Gamma$ with a sign of $T$ and members of $\Delta$ with a sign of $F$. That is, it is a proof in Smullyan's system.

In the other direction, suppose $\mathcal{T}$ is a tableau proof of a sequent $\Gamma \longrightarrow \Delta$ in Smullyan's system. Then $\mathcal{T}$ will also be a correctly constructed tableau in our $\mathcal{T}$ system, except that the initial tableau does not correspond to its being a proof of the sequent since members of $\Delta$ don't have the appropriate sign. Throughout $\mathcal{T}$ replace all occurrences of $F$ with occurrences of $\overline{T}$; call the resulting tableau $\mathcal{T}^*$. The result is a correctly constructed tableau where the initial setup is appropriate for $\Gamma \longrightarrow \Delta$ in our $\mathsf{CL}$ system. Further, $R(\mathcal{T}^*) = \mathcal{T}$, so using the Proposition, $\mathcal{T}^*$ must be a $\mathsf{CL}$ tableau proof of $\Gamma \longrightarrow \Delta$.

One final remark concerning $\mathsf{CL}$. Looking at Definition 2.14 and taking the remarks above into account, we see that for $\mathsf{CL}$ the asymmetric logic and the symmetric one coincide. Thus, not surprisingly, $\mathsf{CL}$ is all we can get from two truth values.

## 3.2 Three-Valued Logics

There is only one three-valued Morgan lattice, consisting of a bottom, a top, and a value in between. We use the classical $f$ and $t$ for the bottom and top. The third value is sometimes denoted $\bot$ (generally for $\mathsf{K}_3$), or $\top$ (appropriate for $\mathsf{LP}$), or $\frac{1}{2}$. We will use $m$ (for middle). Then throughout the three-valued discussion, $M = \{f, m, t\}$, and $\leq$ is as given in Figure 3.5a. The De Morgan involution amounts to turning this figure upside down. The corresponding truth tables appear in Figure 3.5b.

$$\begin{tikzcd}
t \arrow[d, dash] \\
m \arrow[d, dash] \\
f
\end{tikzcd}$$

(a) Three Truth Values

| $\wedge$ | $f$ | $m$ | $t$ |
|---|---|---|---|
| $f$ | $f$ | $f$ | $f$ |
| $m$ | $f$ | $m$ | $m$ |
| $t$ | $f$ | $m$ | $t$ |

| $\vee$ | $f$ | $m$ | $t$ |
|---|---|---|---|
| $f$ | $f$ | $m$ | $t$ |
| $m$ | $m$ | $m$ | $t$ |
| $t$ | $t$ | $t$ | $t$ |

| $\supset$ | $f$ | $m$ | $t$ |
|---|---|---|---|
| $f$ | $t$ | $t$ | $t$ |
| $m$ | $m$ | $m$ | $t$ |
| $t$ | $f$ | $m$ | $t$ |

| $\neg$ | |
|---|---|
| $f$ | $t$ |
| $m$ | $m$ |
| $t$ | $f$ |

(b) Three-Valued Truth Tables

Figure 3.5: Three-Valued Semantics Basics

A three-valued valuation is now is a mapping from formulas to $\{f, m, t\}$ that makes the syntactical connectives and the operations from Subfigure 3.5b correspond, that is, it meets the conditions of Definition 2.8. We remind the reader that a valuation is completely determined by its behavior on propositional letters.

## Symmetric Three-Valued Logics

In the two-valued setting there is just one proper prime filter. In the three-valued Morgan lattice there are two. These give rise to the familiar logics $\mathsf{K}_3$, Kleene's strong three valued logic (from [21]), and $\mathsf{LP}$, Priest's logic of paradox (originating in [3]). We use $\mathsf{K}_3$ and $\mathsf{LP}$ for these particular three valued logics, and also for the algebraic structures given here that determine them. These structures are not unique in the sense that many structures determine the same logics, but when referring to algebraic structures we use the notation only for those discussed here.

**Definition 3.3 (Designated Values)** *The prime filters, and thus the sets of designated values are:*

1. *for* $\mathsf{K}_3$ *the set* $\{t\}$*;*

2. *for* $\mathsf{LP}$ *the set* $\{m, t\}$.

*Validity for formulas and for sequents is as in Definition 2.9.*

For tableaus now, all four signs are essential; no reduction such as for $\mathsf{CL}$ is possible. How tableau signs are interpreted is specified generally in Definition 2.11, and for the current logics this gives us the following table.

**Definition 3.4 (Sign Interpretation)** *The truth value sets assigned for* $\mathsf{K}_3$ *and* $\mathsf{LP}$ *are as follows.*

| *Sign* | $\mathsf{K}_3$ | $\mathsf{LP}$ |
|---|---|---|
| $[\![T]\!]$ | $\{t\}$ | $\{m,t\}$ |
| $[\![\overline{T}]\!]$ | $\{f,m\}$ | $\{f\}$ |
| $[\![F]\!]$ | $\{f\}$ | $\{f,m\}$ |
| $[\![\overline{F}]\!]$ | $\{m,t\}$ | $\{t\}$ |

We have the full set of tableau rules from Definition 2.18 (in uniform notation), or from Figure 2.1 (unabbreviated). Starting a tableau proof is covered by Definition 2.13. What it means to be a closed tableau is given in Definition 2.15, and this yields the following for our present cases.

**Definition 3.5 (Three-Valued Closure)** *For both* $\mathsf{K}_3$ *and* $\mathsf{LP}$*, a tableau branch is closed if the branch contains one of the following pairs of formulas:*

1. $T\,X$ *and* $\overline{T}\,X$*,*
2. $F\,X$ *and* $\overline{F}\,X$*.*

*In addition, a branch is closed if:*

3. *the logic is* $\mathsf{K}_3$*, and the branch contains* $T\,X$ *and* $F\,X$*,*
4. *the logic is* $\mathsf{LP}$*, and the branch contains* $\overline{T}\,X$ *and* $\overline{F}\,X$*.*

Figure 3.6 shows a tableau for the sequent $X \supset Y \longrightarrow (X \supset \neg Y) \supset \neg X$. It is, in fact, simultaneously a *closed* $\mathsf{LP}$ tableau and a $\mathsf{K}_3$ tableau that is *not closed*. Here are the details.

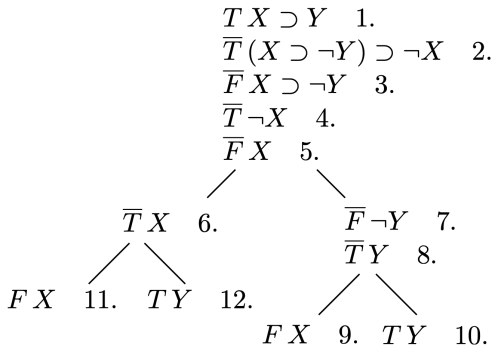

$T\,X \supset Y$ 1.
$\overline{T}\,(X \supset \neg Y) \supset \neg X$ 2.
$\overline{F}\,X \supset \neg Y$ 3.
$\overline{T}\,\neg X$ 4.
$\overline{F}\,X$ 5.
$\overline{T}\,X$ 6.
$\overline{F}\,\neg Y$ 7.
$\overline{T}\,Y$ 8.
$F\,X$ 11.
$T\,Y$ 12.
$F\,X$ 9.
$T\,Y$ 10.

Figure 3.6: $\mathsf{LP}$ and $\mathsf{K}_3$ Tableau

The tableau begins with the initial conditions from Definition 2.13. This gets us items 1 and 2. Then the branch extension rules from Definition 2.18 are followed. Item 2 produces items 3 and 4; 4 produces 5; 3 produces 6 and 7; 7 produces 8; 1 produces 9 and 10. Items 11 and 12 have not yet been added—ignore them for the moment. At this point the tableau is closed using the $\mathsf{LP}$ conditions in Definition 3.5. Closure happens because of: 5 and 6; 5 and 9; 8 and 10.

If, instead, we use the $\mathsf{K}_3$ closure conditions from Definition 3.5, the two right-most branches are closed as before, because of 5 and 9, and 8 and 10, but at this point the branch currently ending with 6 is not closed. It is also the case that not all applicable rules have been applied on the branch ending with 6. Specifically, no rule has been applied to 1 on this branch. Applying a rule yields 11 and 12. The branch ending with 11 is closed because of 5 and 11, but the branch ending with 12 remains unclosed. At this point every applicable rule has been applied, and we do not have a $\mathsf{K}_3$ proof.

The $\mathsf{K}_3$ branch ending with 12, in fact, gives us a $\mathsf{K}_3$ counter-model. The branch contains $T\,Y$ and for $\mathsf{K}_3$, $[\![T]\!] = \{t\}$, so we want a valuation $v$ such that $v(Y) = t$. The branch contains both $\overline{T}\,X$ (6) and $\overline{F}\,X$ (5). For $\mathsf{K}_3$ we have $[\![\overline{T}]\!] = \{f, m\}$ and $[\![\overline{F}]\!] = \{m, t\}$, and these overlap on $m$. So, we want a valuation $v$ such that $v(X) = m$. It is easy to check that our valuation $v$ is a countermodel, in $\mathsf{K}_3$, to $X \supset Y \longrightarrow (X \supset \neg Y) \supset \neg X$.

## Asymmetric Three-Valued Logics

In recent years there has been much interest in what we are calling *asymmetric* three-valued logics. These use not one but two designated sets, a *strict* one and a weaker *tolerant* one. In terms of sequents, the two sides of the arrow are not held to the same standards in an asymmetric logic. Two such logics are common. One is $\mathsf{ST}$, *strict-tolerant* logic, which uses the designated set of $\mathsf{K}_3$ for the left of the arrow and the designated set of $\mathsf{LP}$ for the right (note that the first is a subset of the second, hence stricter). The second asymmetric logic is $\mathsf{TS}$, *tolerant-strict* logic, which reverses the designated set roles.

It is easy to represent the two different designated sets of $\mathsf{ST}$ and $\mathsf{TS}$ using the kind of machinery we have. According to Proposition 2.6, if $D$ is a prime filter in a Morgen lattice, then $\overline{\neg D}$, or equivalently $\neg\overline{D}$, is another. Using the three-valued lattice from Figure 3.5a, if we take $D$ to be the prime filter $\{t\}$, the designated set for $\mathsf{K}_3$, then $\overline{\neg D}$ is $\{m, t\}$, the designated set for $\mathsf{LP}$, and similarly if $D$ is $\{m, t\}$, the designated set for $\mathsf{LP}$, then $\overline{\neg D}$ is $\{t\}$, the designated set for $\mathsf{K}_3$. Either way, we have the strict and tolerant designated sets of $\mathsf{ST}$ and $\mathsf{TS}$. Then we could easily build both asymmetric logics starting with either $\mathsf{K}_3$ or with $\mathsf{LP}$.

Rather than choose just one of $\mathsf{LP}$ or $\mathsf{K}_3$ as the basis for both $\mathsf{ST}$ and $\mathsf{TS}$, we simply make use of both by working with *asymmetric* validation, Definition 2.9. If we choose a base of $\mathsf{K}_3$ we have that $[\![T]\!] = \{t\}$ and $[\![\overline{F}]\!] = \{m, t\}$ and we have $\mathsf{ST}$. If we choose a base of $\mathsf{LP}$ we have that $[\![T]\!] = \{m, t\}$ and $[\![\overline{F}]\!] = \{t\}$ and we have $\mathsf{TS}$.

Now we start the formal tableau development for $\mathsf{ST}$. Following the discussion above, the truth value sets associated with the tableau signs are in the $\mathsf{K}_3$ column of Definition 3.4; the strictly designated set for $\mathsf{ST}$ is $[\![T]\!] = \{t\}$, and the weakly designated set is $[\![\overline{F}]\!] = \{m, t\}$, both prime filters. Then Definition 2.14 specifies how asymmetric tableaus for $\mathsf{ST}$ should start. An $\mathsf{ST}$ tableau proof of the sequent $\Gamma \longrightarrow \Delta$ should start with signed formulas $T\,X_1, \ldots, T\,X_n, F\,Y_1, \ldots, F\,Y_k$ where each $X_i \in \Gamma$ and each $Y_j \in \Delta$. The tableau branch extension rules are, as usual, those in Definition 2.18. Since we are building on the tableau system for $\mathsf{K}_3$, the closure conditions are items 1, 2, and 3 from Definition 3.5.

The logic $\mathsf{TS}$ is like $\mathsf{ST}$ but with roles switched around, so we use the tableau signs from the $\mathsf{LP}$ column of Definition 3.4. Then $[\![T]\!] = \{m, t\}$ and $[\![\overline{F}]\!] = \{t\}$. Pretty much everything concerning tableaus is the same as it was for $\mathsf{ST}$. The conditions for an initial tableau are exactly as before. The branch extension rules are the same as before. Where things differ is in the rules for branch closure; they are now the rules for $\mathsf{LP}$. That is, the closure rules for $\mathsf{TS}$ are items 1, 2, and 4 from Definition 3.5.

In short, as we formulate them $\mathsf{ST}$ and $\mathsf{TS}$ have the same tableau starting conditions, and the same construction rules. They differ only in their closure conditions. Rather than specific tableau examples, we discuss some general observations. The contents are well-known, but are particularly easy to see in a tableau context.

First, note that an ST or TS tableau for a sequent must begin using only the signs $T$ and $F$. The tableau rules all turn $T$ and $F$ signed formulas into other $T$ and $F$ signed formulas. But then only $T$ and $F$ signed formulas can be present in an ST or TS tableau—no occurrences of $\overline{T}$ or $\overline{F}$. So closure conditions 1 and 2 from Definition 3.5 can never be applicable in an ST or TS tableau.

For ST tableaus, only closure condition 3 is significant. But then the proof system is really the same as the standard Smullyan one for classical logic, or equivalently the classical four sign version for classical logic, reduced to two signs as described in Section 3.1. Briefly, ST has the same provable sequents that classical logic does, and thus has the same validities.

For TS we only have closure condition 4 left. But this requires that a signed formula with an overbar, $\overline{T}$ or $\overline{F}$, be involved, and such signed formulas cannot appear in a TS tableau. Our tableau system for TS cannot prove anything! As a matter of fact, it is well known that TS has no valid sequents. Thus the failure of our tableau system to prove anything is exactly what should have happened.

So, ST and CL agree on their valid/provable sequents, while TS has no valid/provable sequents at all. This is far from saying ST and TS have no interest; indeed, the literature concerning them is large. Most of it is not relevant to our present concerns, but there is one item that we can easily address. In [27] attention was fruitfully drawn to a notion dual to validity, *anti-validity*, and in [8] anti-validity was examined in depth. We connected anti-validity with tableaus in Section 2.6, specifically in Propositions 2.22 and 2.23. We now apply this to the logics ST and TS. From Proposition 2.23, anti-validity in ST and in TS is shown for a sequent $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$ by producing separate closed tableaus for each $\overline{T}\,X_i$ and for each $\overline{F}\,Y_j$. Once again the difference between ST and TS comes in the closure rules. These were given in Definition 3.5 and are restated in Figure 3.7 for convenience.

There are some easy to spot issues. A tableau beginning with a formula signed by $\overline{T}$ or $\overline{F}$ can only contain signed formulas whose sign is $\overline{T}$ or $\overline{F}$. (see Figure 2.1). Then for both systems ST and TS the closure condition involving $T$ and $\overline{T}$, and the closure condition involving $F$ and $\overline{F}$ can never be applied, and can be dropped. Likewise the third closure condition for ST can be dropped since it involves only $T$ and $F$. In short, ST has no anti-validities.

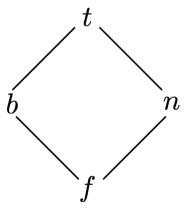
| $\mathsf{K}_3$ and ST | LP and TS |
|---|---|
| $T\,X$ and $\overline{T}\,X$ | $T\,X$ and $\overline{T}\,X$ |
| $F\,X$ and $\overline{F}\,X$ | $F\,X$ and $\overline{F}\,X$ |
| $T\,X$ and $F\,X$ | $\overline{T}\,X$ and $\overline{F}\,X$ |

Figure 3.7: Tableau Branch Closure Conditions

For TS, closure involving $\overline{T}$ and $\overline{F}$ is possible. But then, in brief, a TS tableau verifying anti-validity, starts with, continues with, and closes with only $\overline{T}$ and $\overline{F}$ signs. Now, at the end of Section 3.1 we discussed the replacement of $\overline{T}$ with $F$ and $\overline{F}$ with $T$, and saw that it mapped closed tableaus into closed tableaus, but they are tableaus in Smullyan's two-sign system. Applying that mapping here, we see that the anti-validities of TS are the same as the antivalidities of CL.

## 3.3 Four-Valued Logic

We have discussed tableau systems for four different three valued logics, two symmetric and two asymmetric. With four truth values arranged in the familiar diamond configuration the possibilities are even more limited. Essentially there is only one logic, but it is a famous one: *first degree* *entailment*, or FDE. In a sense the truth value structure for FDE combines the ones for $\mathsf{K}_3$ and LP. The two are different because of the intended behavior of the middle value $m$, designated or not. The FDE structure simply incorporates both versions. We use $n$ and $b$ for the two middle values, standing for *neither true nor false* and *both true and false* respectively. Figure 3.8a displays the standard structure for FDE. The ordering is upward, $b \leq t$ and so on. As usual here, conjunction is interpreted as meet, disjunction as join, De Morgan involution, negation, is vertical symmetry, switching $t$ and $f$, leaving $b$ and $n$ alone, and implication is defined from the other connectives. The set of designated truth values is taken to be $\{t, b\}$. Truth tables for the operations are shown in Figure 3.8b.



(a) Four Truth Values

| $\wedge$ | $f$ | $b$ | $n$ | $t$ |
|---|---|---|---|---|
| $f$ | $f$ | $f$ | $f$ | $f$ |
| $b$ | $f$ | $b$ | $f$ | $b$ |
| $n$ | $f$ | $f$ | $n$ | $n$ |
| $t$ | $f$ | $b$ | $n$ | $t$ |

| $\vee$ | $f$ | $b$ | $n$ | $t$ |
|---|---|---|---|---|
| $f$ | $f$ | $b$ | $n$ | $t$ |
| $b$ | $b$ | $b$ | $t$ | $t$ |
| $n$ | $n$ | $t$ | $n$ | $t$ |
| $t$ | $t$ | $t$ | $t$ | $t$ |

| $\supset$ | $f$ | $b$ | $n$ | $t$ |
|---|---|---|---|---|
| $f$ | $t$ | $t$ | $t$ | $t$ |
| $b$ | $b$ | $b$ | $t$ | $t$ |
| $n$ | $n$ | $t$ | $n$ | $t$ |
| $t$ | $f$ | $b$ | $n$ | $t$ |

| $\neg$ | |
|---|---|
| $f$ | $t$ |
| $b$ | $b$ |
| $n$ | $n$ |
| $t$ | $f$ |

(b) Four-Valued Truth Tables

Figure 3.8: Four-Valued Semantics Basics

Valuations now map to the FDE structure, and meet the conditions of Definition 2.8. Since our designated set is now $\{t, b\}$, Definition 2.11 gives us the following interpretation of tableau signs.

$$
\begin{aligned}
[\![T]\!] &= \{t, b\} \\
[\![F]\!] &= \{f, b\} \\
[\![\overline{T}]\!] &= \{f, n\} \\
[\![\overline{F}]\!] &= \{t, n\}
\end{aligned}
$$

Since the value $b$ is commonly understood as *both* true and false, the sign $T$ can be read as *at least true*: either just true, or true and also false. Likewise $F$ is *at least false.* Then $\overline{T}$ and $\overline{F}$ can be read as *at most false* and *at most true.*

Initial tableaus are from Definitions 2.13 and 2.14. Branch extension rules are from Definition 2.18 or as expanded, from Figure 2.1. The tableau closure conditions are determined by Definition 2.15. They work out to the following.

**Definition 3.6 (Four-Valued Closure)** *For the four-valued structure, a tableau branch is (atomically) closed if, for some (atomic) formula $X$, the branch contains one of the following pairs of formulas:*

1. *$T\,X$ and $\overline{T}\,X$,*
2. *$F\,X$ and $\overline{F}\,X$,*

The tableau system that results from all this first appeared in [7]. Figure 3.9 shows an example of an FDE proof. This example is taken from [26], where it is proved using the rules for FDE tableaus as given in that book. Thus the two proof systems can be compared. The proof is for the sequent $\neg(B \wedge \neg C) \wedge A \longrightarrow (\neg B \vee C) \vee D$. We note that the sequent involved in Figure 3.6 is not provable in FDE, though it was in LP. It is a good exercise to verify this.

$$\begin{array}{l}
T\,\neg(B \wedge \neg C) \wedge A \quad 1. \\
\overline{T}\,(\neg B \vee C) \vee D \quad 2. \\
T\,\neg(B \wedge \neg C) \quad 3. \\
T\,A \quad 4. \\
F\,B \wedge \neg C \quad 5. \\
\overline{T}\,\neg B \vee C \quad 6. \\
\overline{T}\,D \quad 7. \\
\overline{T}\,\neg B \quad 8. \\
\overline{T}\,C \quad 9. \\
\overline{F}\,B \quad 10.
\end{array}$$

$$\begin{array}{ll}
F\,B \quad 11. & F\,\neg C \quad 12. \\
 & T\,C \quad 13.
\end{array}$$

Figure 3.9: FDE Proof Example

There is really no asymmetric version of FDE. An asymmetric initial tableau, Definition 2.14, has only $T$ and $F$ as signs. A look at the branch extension rules from Figure 2.1 shows that $\overline{T}$ and $\overline{F}$ can never turn up in a subsequent tableau, and hence the closure conditions from Definition 3.6 can never be applied.

## 3.4 Six-Valued Logic

We have been using the terminology "Morgan lattice", and not the more common "De Morgan lattice". The latter requires distributivity, but we have not needed it. We now give a specific example of a non-distributive Morgan lattice, along with a prime filter, and see what logic it determines. The lattice is displayed in Figure 3.10[1]. As usual here, the lattice ordering is upward, and the Morgan involution simply inverts the structure.

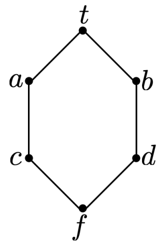


Figure 3.10: A Non-Distributive Lattice

It is easy to check that this is a Morgan lattice, but it is not distributive. For instance, $a \wedge (b \vee c) = a \wedge t = a$, but $(a \wedge b) \vee (a \wedge c) = f \vee c = c$. Thus it is not a De Morgan lattice.

[1]I thank Hanamantagouda Sankappanavar for suggesting this example.

The subset $\{c, a, t\}$ is a proper prime filter. (So is $\{d, b, t\}$, but it is a symmetric counterpart of the first, and has similar behavior. We do not discuss it further.) Using this prime filter as our designated set we have the following.

$$\begin{aligned}
[\![T]\!] &= \{c, a, t\}\\
[\![F]\!] &= \{c, a, f\}\\
[\![\overline{T}]\!] &= \{b, d, f\}\\
[\![\overline{F}]\!] &= \{b, d, t\}
\end{aligned}$$

The only pairs that do not have a non-empty overlap are: $T$, $\overline{T}$ and $F$, $\overline{F}$. Thus the closure rules we get are actually those of FDE, and hence that is the logic being determined.

## 3.5 One More Thing, S3

There is a less conventional logic, S3, that can be handled by our tableau machinery with a little tweaking. Its name was introduced in [12] to stand for "symmetric 3-valued", but the logic itself is older. It is not a standard many-valued logic directly, but lives in the general vicinity. A sequent is considered valid in S3 if it is valid in *both* $K_3$ and LP, thus it is an intersection logic. We sketch how our tableau systems can handle S3, but leave most of the details to the reader.

Tableaus for all the semantically defined symmetric logics we considered start the same way. A tableau to establish that $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$ is a validity begins construction with the signed formulas $T\,X_1, \ldots, T\,X_n, \overline{T}\,Y_1, \ldots, \overline{T}\,Y_k$. They all continue using the same branch extension rules, given in Figure 2.1. Systems differ in their branch closure conditions, and the ones for $K_3$ and LP are summarized in Figure 3.7. Then for S3, we simply require that each tableau branch be closed using *both* the $K_3$ and the LP conditions.

For example, the tableau for $X \supset Y \longrightarrow (X \supset \neg Y) \supset \neg Y$ shown in Figure 3.6, is closed in the LP sense but not in the $K_3$ sense. Indeed, a $K_3$ counter-model was extracted from the tableau. So the sequent has been established to *not* be an S3 validity. On the other hand, consider the sequent $A \wedge \neg A \longrightarrow B \vee \neg B$, which is characteristic for S3. Figure 3.11 displays a tableau for it, with all applicable rules actually applied. The only branch is closed using the $K_3$ rules because of 3 and 5. It is also closed using the LP rules because of 6 and 8. This establishes that the sequent is S3 valid.

$$\begin{array}{ll}
T\,A \wedge \neg A & 1.\\
\overline{T}\,B \vee \neg B & 2.\\
T\,A & 3.\\
T\,\neg A & 4.\\
F\,A & 5.\\
\overline{T}\,B & 6.\\
\overline{T}\,\neg B & 7.\\
\overline{F}\,B & 8.
\end{array}$$

Figure 3.11: S3 Tableau Proof

One final remark about S3 and tableaus. The closure rules for FDE from Definition 3.6 are closure conditions for both $K_3$ and LP. It follows that the validities of FDE are among those of S3. This is, of course, well known. The present point is simply that this is an easy consequence of the tableau formulations.

## 4 There Isn't Much To Go Around

Each prime logical Morgan lattice $\langle \mathsf{M}, D\rangle$ semantically determines a many-valued logic, using its prime filter $D$ as the set of designated truth values. For each such logic there is a corresponding tableau system. All these tableau systems have the same initial conditions, Definition 2.13, and the same branch extension rules, Definition 2.18 and Figure 2.1. Closure conditions are determined by Definition 2.15, so any differences between two such logics must arise because of different closure rules. Figure 4.1 collects together the ones we have seen. For every logic arising from a prime logical Morgan lattice we must always have $T, \overline{T}$, and $F, \overline{F}$ among the closure conditions, since these cases involve complementation and so there can be no overlap. This leaves only a small number of other potential closure conditions, four in fact: $T$, $F$ or $T$, $\overline{F}$ or $\overline{T}$, $F$ or $\overline{T}$, $\overline{F}$. In fact, two of these are not possible: $T$, $\overline{F}$ and $\overline{T}$, $F$.

Here is the argument for why the case $T$, $\overline{F}$ cannot be a closure condition. For it to be one, $[\![T]\!]$ and $[\![\overline{F}]\!]$ must not overlap. By Definition 2.11 this says that $D$ and $\overline{\neg D}$ cannot overlap, from which it follows that $D \subseteq \neg D$. Now $D$, being a non-empty filter, is upward closed and hence contains $t$, the top member of the lattice. Then $t \in \neg D$, so $\neg t \in D$, that is, $f \in D$. Since $D$ is upward closed, everything is in $D$, so it is not a proper filter. A similar argument works for the case $\overline{T}$, $F$.

It follows that the combinations of closure conditions shown in Figure 4.1 are the only ones possible, and so while there are infinitely many prime logical Morgan lattices, there are only four symmetric logics that any of them can determine. And those four are already represented among the Morgan lattices with at most four members. Adding asymmetric logics changes the game, but it only adds a few more cases to the symmetric ones.

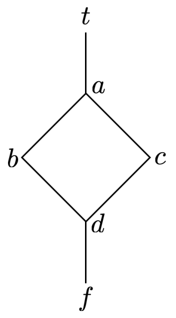
| Logic | Closure Conditions |
|---|---|
| $\mathsf{CL}$ | $T, \overline{T}$ or $F, \overline{F}$ or $T, F$ or $\overline{T}, \overline{F}$ |
| $\mathsf{K}_3$ | $T, \overline{T}$ or $F, \overline{F}$ or $T, F$ |
| $\mathsf{LP}$ | $T, \overline{T}$ or $F, \overline{F}$ or $\overline{T}, \overline{F}$ |
| $\mathsf{FDE}$ | $T, \overline{T}$ or $F, \overline{F}$ |

Figure 4.1: Possible Symmetric Branch Closure Conditions

We conclude this section with one more De Morgan lattice example, displayed in Figure 4.2. It illustrates the use of tableaus to determine what logic an example presents. Figure 4.2a shows a rather standard six element lattice. It has four proper prime filters: $D_1 = \{t\}$, $D_2 = \{t, a, c\}$, $D_3 = \{t, a, b\}$, and $D_4 = \{t, a, b, c, d\}$. Since $D_2$ and $D_3$ are symmetric versions of each other, we omit $D_3$ from further discussion. For the rest, in Figure 4.2b we show the results of applying Definition 2.11, and say what the closure conditions are, as determined by Definition 2.15. The logics we get are $\mathsf{K}_3$, $\mathsf{FDE}$, and $\mathsf{LP}$.

## 5 Interpolation

This section is devoted to showing that interpolation can be proved rather easily using the tableau rules of this paper, and done so uniformly for all symmetric logics in the family determined by prime logical Morgan lattices (which really comes down to four logics). Interpolation for first degree entailment was proved in [1]. Interpolation for $\mathsf{K}_3$ can be found in [6, 22]. A uniform proof related to ours can be found in [29]. Our proof ultimately traces back to an elegant constructive one of Smullyan using tableaus for classical logic, found in [28]. Later this method was used in [17]



(a) De Morgan Lattice

| | $D_1$ | $D_2$ | $D_4$ |
|---|---|---|---|
| $[\![T]\!]$ | $\{t\}$ | $\{t,a,c\}$ | $\{t,a,b,c,d\}$ |
| $[\![\overline{T}]\!]$ | $\{a,b,c,d,f\}$ | $\{b,d,f\}$ | $\{f\}$ |
| $[\![F]\!]$ | $\{f\}$ | $\{c,d,f\}$ | $\{f,a,b,c,d\}$ |
| $[\![\overline{F}]\!]$ | $\{t,a,b,c,d\}$ | $\{t,a,b\}$ | $\{t\}$ |
| Conditions | $T,\overline{T}$ | $T,\overline{T}$ | $T,\overline{T}$ |
| | $F,\overline{F}$ | $F,\overline{F}$ | $F,\overline{F}$ |
| | $T,F$ | | $\overline{T},\overline{F}$ |
| Logics | $\mathsf{K}_3$ | FDE | LP |

(b) Tableau Closure Conditions

Figure 4.2: A Six-Valued Example

to prove interpolation for several standard modal logics such as K4 and S4, with an extension to Gödel-Löb logic in [14]. Smullyan's formal machinery involved a special two-sided sequent calculus. In [15] different notational machinery was introduced, involving what we called *biased formulas.* It turns out that the present structure of four signs provides us with a direct equivalent of biasing, and so Smullyan's methodology applies with no additional formal machinery needed.

The original interpolation theorem was for classical logic, [9], and says that if $X \supset Y$ is classically valid then there is an interpolant, $Z$, in the common language for $X$ and $Y$, such that both $X \supset Z$ and $Z \supset Y$ are classically valid. Propositionally, "common language" means that all propositional letters of $Z$ are common to $X$ and $Y$. Of course there is the issue of what to do if $X$ and $Y$ have no propositional letters in common. Classically, if this happens then either $\neg X$ is valid or $Y$ is valid. However things can be stated more simply if we add propositional constants to the language, $\bot$ and $\top$, interpreted as classical falsity and truth respectively. If this is done, then the new symbols are always part of the common language, and there are no special cases that need to be considered. We follow this route.

**Special Assumption** For this section only, our propositional language is expanded with $\bot$ and $\top$, and Definition 2.8 is extended with the following. In each Morgan lattice, where $f$ denotes the bottom element and $t$ denotes the top, we require the following of valuations:

$$v(\bot) = f$$
$$v(\top) = t.$$

Of course with new propositional constants in the language, tableau systems need extending too. The idea behind Definition 2.15 was that a branch should close if it represented an impossible situation, and this required certain pairs of signed formulas to be present. With the new propositional constants, closure can also be because of a single signed formula.

**Definition 5.1 (New Closure Conditions)** *For this section with $\top$ and $\bot$ present, a tableau branch is closed if it meets the appropriate closure conditions from earlier, or if it contains any one of the following signed formulas:*

$$T\,\bot, \quad \overline{F}\,\bot, \quad F\,\top, \quad \overline{T}\,\top.$$

All four cases above represent impossible situations. We check two of them. First, suppose we have $T\,\bot$ on a tableau branch. According to Definition 2.12, if a valuation $v$ were to validate this then $v(\bot) \in [\![T]\!]$, and so $f \in D$. But since $D$ is upward closed, it would contain everything and thus not be a proper filter. Likewise if $v$ validates $\overline{F}\,\bot$ we would have $v(\bot) \in [\![\overline{F}]\!]$, and so $f \in \overline{\neg D}$. But then $f \notin \neg D$, and so $t \notin D$. But again $D$ is upward closed, and it would follow that $D$ is empty, again not a proper filter.

Our soundness and completeness results for tableaus, with respect to prime logical Morgan lattices, extend very easily to admit $\bot$ and $\top$ into the language. In this section we assume such soundness and completeness, and omit the verification.

**Proposition 5.2 (Interpolation)** *Let $\langle \mathsf{M}, D\rangle$ be a prime Morgan lattice, and assume that $\langle \mathsf{M}, D\rangle \models \Gamma \longrightarrow \Delta$. Then there is a formula $Z$, with all its propositional letters common to $\Gamma$ and $\Delta$, such that $\langle \mathsf{M}, D\rangle \models \Gamma \longrightarrow Z$ and $\langle \mathsf{M}, D\rangle \models Z \longrightarrow \Delta$.*

The rest of the section is devoted to a proof of the Proposition. We assume throughout that $\langle \mathsf{M}, D\rangle$ is a prime Morgan lattice. Throughout, the word "tableau" is often used as short for "$\langle \mathsf{M}, D\rangle$ tableau". All the work is really done in a Lemma that refers directly to tableaus, and yields Proposition 5.2 as a special case. We begin with some special terminology and notation. This machinery makes an important distinction between unbarred tableau signs, $T$ and $F$, and barred tableau signs, $\overline{T}$ and $\overline{F}$—this is the counterpart here of biasing in [15].

**Definition 5.3** *We introduce functions that separate sets of signed formulas into two categories. For a set $S$ of signed formulas:*

$$\begin{aligned} \mathit{unbar}(S) &= \{T\,W \mid T\,W \in S\} \cup \{F\,W \mid F\,W \in S\} \\ \mathit{bar}(S) &= \{\overline{T}\,W \mid \overline{T}\,W \in S\} \cup \{\overline{F}\,W \mid \overline{F}\,W \in S\}. \end{aligned}$$

*We say a formula $Z$ is an* interpolant *in $\langle \mathsf{M}, D\rangle$ for a set $S$ of signed formulas if we have the following:*

1. *All propositional letters in $Z$ occur in formulas of both $\mathit{bar}(S)$ and $\mathit{unbar}(S)$,*
2. *There are closed $\langle \mathsf{M}, D\rangle$ tableaus for both $\mathit{unbar}(S) \cup \{\overline{T}\,Z\}$ and $\mathit{bar}(S) \cup \{T\,Z\}$.*

**Lemma 5.4** *If there is a closed $\langle \mathsf{M}, D\rangle$ tableau for a finite set $S$ of signed formulas, then $S$ has an interpolant.*

**Proof** The proof is by induction on the number of tableau rule applications it takes to close a tableau for $S$. For all of the following, assume we have a closed tableau $\mathcal{T}$ for the finite set $S$, and the induction hypothesis is that every set of signed formulas having a closed tableau with fewer steps than $\mathcal{T}$ has an interpolant. In presenting the proof it simplifies things to make some use of uniform notation from Definition 2.16.

**Base Cases** Suppose the tableau $\mathcal{T}$ for $S$ closes using 0 rule applications, that is, a tableau containing just the members of $S$ must already be closed. We show that, under these circumstances, $S$ has an interpolant. The only way a zero step tableau for $S$ can be closed is if $S$ meets one of the appropriate closure conditions from Figure 4.1, or if $S$ contains one of the single signed formula conditions from Definition 5.1. How to handle the various cases is summarized in the following table. Note that of the two-formula conditions, the first two apply for all $\langle \mathsf{M}, D\rangle$

while the second two apply for some choices of $\langle \mathsf{M}, D\rangle$ but not for others. We only discuss three of the cases, and leave the rest to the reader.

| $S$ Cases | interpolant |
|---|---|
| $T\,\bot$ | $\bot$ |
| $\overline{F}\,\bot$ | $\top$ |
| $F\,\top$ | $\bot$ |
| $\overline{T}\,\top$ | $\top$ |
| $T\,P, \overline{T}\,P$ | $P$ |
| $F\,P, \overline{F}\,P$ | $\neg P$ |
| $T\,P, F\,P$ | $\bot$ |
| $\overline{T}\,P, \overline{F}\,P$ | $\top$ |

For the following suppose we have an atomically closed tableau with a single branch, $S$ is the set of signed formulas on it, and $S$ contains the formula(s) shown in one of the cases above.

*Case* $\overline{F}\,\bot$*:* Assume $\overline{F}\,\bot \in S$. We show $\top$ is an interpolant. Trivially all propositional letters in $\top$ are common to *unbar*$(S)$ and *bar*$(S)$. Also trivially there is a closed tableau for *unbar*$(S) \cup \{\overline{T}\,\top\}$ because it contains $\overline{T}\,\top$ and we have Definition 5.1. Finally *bar*$(S) \cup \{T\,\top\}$ is closed because $\overline{F}\,\bot$ is in *bar*$(S)$, and again one of the New Closure Conditions applies.

*Case* $F\,P, \overline{F}\,P$*:* This is a closure condition that applies for all choices of $\mathsf{M}$. Suppose $S$ includes $F\,P$ and $\overline{F}\,P$. We show $\neg P$ is an interpolant.

The only propositional letter in $\neg P$ is $P$, and it occurs in *unbar*$(S)$ since $S$ includes $F\,P$, and it occurs in *bar*$(S)$ since $S$ includes $\overline{F}\,P$.

Next we show we have the needed closed tableaus. For one, we want closure for a tableau starting with *unbar*$(S)$ and $\overline{T}\,\neg P$. But *unbar*$(S)$ contains $F\,P$, and from $\overline{T}\,\neg P$ we can get $\overline{F}\,P$, and so we have closure. For the other, we want closure for a tableau starting with $T\,\neg P$ and the members of *bar*$(S)$. From $T\,\neg P$ we can get $F\,P$ and *bar*$(S)$ contains $\overline{F}\,P$, so again we have closure.

*Case* $T\,P, F\,P$*:* This is a closure condition that is not universal. Assume $\langle \mathsf{M}, D\rangle$ is a prime logical Morgan lattice for which the condition is appropriate, and $S$ contains $T\,P$ and $F\,P$; we show $\bot$ is an interpolant.

Since $\bot$ contains no propositional letters, vacuously all occur in both *bar*$(S)$ and in *unbar*$(S)$.

We need tableau proofs for *unbar*$(S) \cup \{\overline{T}\,\bot\}$ and for *bar*$(S) \cup \{T\,\bot\}$. A tableau proof for *unbar*$(S) \cup \{\overline{T}\,\bot\}$ is immediate since both $T\,P$ and $F\,P$ are in *unbar*$(S)$, and this is subject to our closure condition. A proof for *bar*$(S) \cup \{T\,\bot\}$ is equally immediate, since this contains $T\,\bot$, and that is one of our New Closure Conditions.

**Negation Induction Cases,** $\rho$ Suppose $\rho \in S$ and in the closed tableau $\mathcal{T}$ for $S$ the first rule applied in $\mathcal{T}$ is to it, adding $\rho_0$. The $\rho$ rule from Definition 2.18 summarizes four negation cases. We discuss one of them in detail, conclude $F\,X$ from $T\,\neg X$. The other three are similar and are left to the reader.

Suppose $T\,\neg X \in S$ and in the closed tableau $\mathcal{T}$ for $S$ the first rule applied adds $F\,X$ to the initial branch containing the members of $S$. Instead of considering the tableau $\mathcal{T}$ as one for $S$ with its first rule application being to $T\,\neg X$, we could also understand $\mathcal{T}$ as a tableau for the set $S \cup \{F\,X\}$, and whose first rule application is whatever originally followed the initial

Negation rule application. This is also a closed tableau but with one fewer rule application, so the induction hypothesis applies. The set $S \cup \{F\,X\}$ must have an interpolant, say $Z$. We show $Z$ is also an interpolant for the original set $S$.

Trivially $bar(S \cup \{F\,X\}) = bar(S)$ so, since the propositional letters occurring in $Z$ are in $bar(S \cup \{F\,X\})$, they are in $bar(S)$. Also trivially, $X$ and $\neg X$ contain the same propositional letters. And, since $T\,\neg X \in S$, the propositional letters appearing in $unbar(S \cup \{F\,X\})$ and in $unbar(S)$ are the same. Since the propositional letters occurring in $Z$ are in $unbar(S \cup \{F\,X\})$, they are in $unbar(S)$.

Since $Z$ is an interpolant for $S \cup \{F\,X\}$, there is a closed tableau $\mathcal{T}_1$ beginning with $unbar(S \cup \{F\,X\}) \cup \{\overline{T}\,Z\}$. Equivalently, $\mathcal{T}_1$ is a closed tableau beginning with $unbar(S) \cup \{F\,X, \overline{T}\,Z\}$. Then there is a closed tableau beginning with $unbar(S) \cup \{\overline{T}\,Z\}$ because, since $T\,\neg X \in unbar(S)$, we can start by applying the $T\,\neg$ rule to add $F\,X$, and then copy the steps of $\mathcal{T}_1$. There is also a closed tableau $\mathcal{T}_2$ beginning with $bar(S \cup \{F\,X\}) \cup \{T\,Z\}$. But $\mathcal{T}_2$ is directly a closed tableau for $bar(S) \cup \{T\,Z\}$.

**Non-Branching Induction Cases,** $\alpha$ Suppose $\alpha \in S$ and in the closed tableau $\mathcal{T}$ for $S$ the first rule applied is to $\alpha$, adding $\alpha_1$ and $\alpha_2$ to the initial branch containing the members of $S$. Instead of considering the tableau $\mathcal{T}$ as one for $S$ with its first rule application being to the $\alpha$ formula, we could also understood $\mathcal{T}$ as a tableau for the set $S \cup \{\alpha_1, \alpha_2\}$, and whose first rule application is whatever originally followed the initial $\alpha$ rule application. This is also a closed tableau but with fewer rule applications, so the induction hypothesis applies. The set $S \cup \{\alpha_1, \alpha_2\}$ must have an interpolant, say $Z$. We show $Z$ is also an interpolant for the original set $S$.

There are two possibilities we need to consider, since $\alpha$ might have a sign with an overbar or a sign without an overbar. We only discuss the overbar case; the other is similar and we omit the details for it. Thus we now assume that $S \cup \{\alpha_1, \alpha_2\}$ has interpolant $Z$, and $\alpha$ has one of $\overline{T}$ or $\overline{F}$ as its sign. Note that $\alpha_1$ and $\alpha_2$ must also have signs with overbars, so all of $\alpha, \alpha_1$ and $\alpha_2$ are in $bar(S \cup \{\alpha_1, \alpha_2\})$. Since $Z$ is an interpolant for $S \cup \{\alpha_1, \alpha_2\}$, all propositional letters in $Z$ occur in formulas of $bar(S \cup \{\alpha_1, \alpha_2\})$ and in $unbar(S \cup \{\alpha_1, \alpha_2\})$, and both $unbar(S \cup \{\alpha_1, \alpha_2\}) \cup \{\overline{T}\,Z\}$ and $bar(S \cup \{\alpha_1, \alpha_2\}) \cup \{T\,Z\}$ have closed $\langle \mathsf{M}, D\rangle$ tableaus.

Now, a propositional letter occurrs in $\alpha$ exactly if the propositional letter occurrs in at least one of $\alpha_1$ or $\alpha_2$. Consequently since $\alpha \in S$, all the propositional letters occurring in $Z$ occur in formulas of $bar(S)$ and in formulas of $unbar(S)$. Also, there is a closed tableau for $unbar(S \cup \{\alpha_1, \alpha_2\}) \cup \{\overline{T}\,Z\}$, and from this we can easily construct a closed tableau for $unbar(S) \cup \{\overline{T}\,Z\}$. Simply begin with an application of the $\alpha$ rule, allowing us to add $\alpha_1$ and $\alpha_2$, then proceed as in the closed tableau for $unbar(S \cup \{\alpha_1, \alpha_2\}) \cup \{\overline{T}\,Z\}$. In a similar way there is a closed tableau for $bar(S) \cup \{T\,Z\}$.

**Branching Induction Cases,** $\beta$ This time suppose $\beta \in S$, $\mathcal{T}$ is a closed tableau for $S$, the induction hypothesis holds for sets with closed tableaus having fewer steps than $\mathcal{T}$, and the first tableau rule applied in $\mathcal{T}$ is to $\beta$, causing the tableau to branch, with a left branch starting with $\beta_1$ and a right branch starting with $\beta_2$. If we cut off the right branch from $\mathcal{T}$ what remains is a properly constructed and closed tableau, call it $\mathcal{T}_1$, but beginning with the set $S \cup \{\beta_1\}$. Similarly if we cut off the left branch from $\mathcal{T}$ we have a closed tableau, $\mathcal{T}_2$, beginning with $S \cup \{\beta_2\}$. Both $\mathcal{T}_1$ and $\mathcal{T}_2$ have fewer steps than $\mathcal{T}$ so the induction hypothesis applies. Thus there is an interpolant, $Z_1$ for the set $S \cup \{\beta_1\}$ and an interpolant $Z_2$ for the set $S \cup \{\beta_2\}$. To sumarize, we have the following.

1. All propositional letters of $Z_1$ occur in formulas of $\mathit{bar}(S \cup \{\beta_1\})$ and of $\mathit{unbar}(S \cup \{\beta_1\})$.
2. There are closed tableaus for both $\mathit{unbar}(S \cup \{\beta_1\}) \cup \{\overline{T}\, Z_1\}$ and $\mathit{bar}(S \cup \{\beta_1\}) \cup \{T\, Z_1\}$.
3. All propositional letters of $Z_2$ occur in formulas of $\mathit{bar}(S \cup \{\beta_2\})$ and of $\mathit{unbar}(S \cup \{\beta_2\})$.
4. There are closed tableaus for both $\mathit{unbar}(S \cup \{\beta_2\}) \cup \{\overline{T}\, Z_2\}$ and $\mathit{bar}(S \cup \{\beta_2\}) \cup \{T\, Z_2\}$.

We will use $Z_1$ and $Z_2$ to construct an interpolant for $S$, but there are two subcases depending on whether $\beta$ has a sign with an overbar or a sign without an overbar. We only discuss the case with no overbar, that is, we assume $\beta$ has either $T$ or $F$ as its sign. The case with an overbar proceeds dually to the present one, and is left to the reader. We will show that $Z_1 \vee Z_2$ is an interpolant for $S$. In the dual case, which we have left to the reader, $Z_1 \wedge Z_2$ is the interpolant.

Any propositional letter occurring in $Z_1 \vee Z_2$ occurs in $Z_1$ or in $Z_2$ (possibly both). If a letter occurs in $Z_1$, by item 1 above it occurs in a formula of $\mathit{bar}(S \cup \{\beta_1\})$ and in a formula of $\mathit{unbar}(S \cup \{\beta_1\})$. But all propositional letters occurring in $\beta_1$ also occur in $\beta$, so any propositional letter occurring in $Z_1$ must occur in both $\mathit{bar}(S)$ and in $\mathit{unbar}(S)$. Similarly for propositional letters in $Z_2$, using item 3. So, every propositional letter in $Z_1 \vee Z_2$ occurs in both $\mathit{bar}(S)$ and in $\mathit{unbar}(S)$.

We must show that there are closed tableaus for both $\mathit{unbar}(S) \cup \{\overline{T}\, Z_1 \vee Z_2\}$ and $\mathit{bar}(S) \cup \{T\, Z_1 \vee Z_2\}$. The second of these is easy. Since we are discussing the case where $\beta$ has either $T$ or $F$ as its sign, $\beta \notin \mathit{bar}(S)$. Likewise $\beta_1$ and $\beta_2$ must have $T$ or $F$ as their sign, so $\mathit{bar}(S \cup \{\beta_1\})$ and $\mathit{bar}(S \cup \{\beta_2\})$ both are just $\mathit{bar}(S)$. Then to get a closed tableau for $\mathit{bar}(S) \cup \{T\, Z_1 \vee Z_2\}$ apply the tableau rule for $T\vee$, splitting to two branches, one with $\mathit{bar}(S) \cup \{T\, Z_1 \vee Z_2, T\, Z_1\}$ and the other with $\mathit{bar}(S) \cup \{T\, Z_1 \vee Z_2, T\, Z_2\}$. This gives us branches that can be continued to closure using items 2 and 4 above.

To complete the argument we need to show there is a closed tableau for $\mathit{unbar}(S) \cup \{\overline{T}\, Z_1 \vee Z_2\}$. Here is the tableau construction. Begin with a single branch containing the members of $\mathit{unbar}(S) \cup \{\overline{T}\, Z_1 \vee Z_2\}$. Applying the $\overline{T}\vee$ rule, add both $\overline{T}\, Z_1$ and $\overline{T}\, Z_2$ to the branch end, and then apply a rule to $\beta$, splitting the branch end with one fork containing $\beta_1$ and the other $\beta_2$. The branch ending in $\beta_1$ can be closed by item 2 above, and the branch ending in $\beta_2$ can be closed by item 4.

■

Now we are ready to prove interpolation in a uniform way for our logics.

**Proof** *of Proposition 5.2* Suppose $\langle \mathsf{M}, D\rangle \models \Gamma \longrightarrow \Delta$. Using our soundness and completeness theorems there are $X_1, \ldots, X_n \in \Gamma$ and $Y_1, \ldots, Y_k \in \Delta$ such that some $\langle \mathsf{M}, D\rangle$ tableau beginning with

$$S = \{T\, X_1, \ldots, T\, X_n, \overline{T}\, Y_1, \ldots, \overline{T}\, Y_k\} \qquad (\star)$$

is closed. By Lemma 5.4 the set $S$ has an interpolant, $Z$. Then all propositional letters in $Z$ are in $\mathit{unbar}(S)$, and hence in members of $\{X_1, \ldots, X_n\} \subseteq \Gamma$. Similarly all propositional letters in $Z$ are in $\mathit{bar}(S)$, hence in members of $\{Y_1, \ldots, Y_k\} \subseteq \Delta$. Further there are closed $\langle \mathsf{M}, D\rangle$ tableaus for $\mathit{unbar}(S) \cup \{\overline{T}\, Z\}$ and for $\mathit{bar}(S) \cup \{T\, Z\}$ and so, using soundness and completeness again, $\langle \mathsf{M}, D\rangle \models \Gamma \longrightarrow Z$ and $\langle \mathsf{M}, D\rangle \models Z \longrightarrow \Delta$. ■

## 6 A Brief History

It would be surprising if the kind of tableau machinery discussed here were entirely new. It is not. What is new is the general algebraic approach. This makes it possible to consider an infinite family of structures, prove things uniformly, interpolation for instance, and ultimately determine that only a small number of logics are actually characterized by an infinite family of natural structures. This section mentions some of the earlier literature. There are several tableau style proof systems that have appeared for FDE, and some for LP. We only bring up those that are closely related to our particular tableau style approach.

A four-sign tableau system for FDE can be found in [10, Section 5]. ($\mathsf{K}_3$ and LP are not treated.). Suppose we think of the four truth values in the lattice for FDE as Belnap did [4, 5]: only true, only false, both, and neither. Then [10] introduces four tableau signs $t$, $f$, $t^*$, $f^*$, and gives them the following meanings.

$$\begin{aligned}
[\![t]\!] &= \{\text{only true}, \text{both}\}\\
[\![f]\!] &= \{\text{only false}, \text{neither}\}\\
[\![t^*]\!] &= \{\text{only true}, \text{neither}\}\\
[\![f^*]\!] &= \{\text{only false}, \text{both}\}
\end{aligned}$$

Comparing things, these signs correlate with our usage for FDE as follows.

$$\begin{aligned}
[\![t]\!] &= [\![T]\!]\\
[\![f]\!] &= [\![\overline{T}]\!]\\
[\![t^*]\!] &= [\![\overline{F}]\!]\\
[\![f^*]\!] &= [\![F]\!]
\end{aligned}$$

Using this correlation, the tableau system in [10] is the same as the symmetric system here for FDE. It would have been possible at the time, of course, to consider extending the rules in [10] to a broader range of algebraic structures beyond FDE, but the work there went in a different direction.

One of the directions investigated in [10] should be mentioned here. Tableaus have a natural application to automated theorem proving. Standard tableau rules are decompositional. A signed formula generates simpler signed formulas. A tableau system called KE considers a different sort of rule, in a sense generalizing *modus ponens*. For instance for a standard classical tableau system it would allow the addition of $T\,Y$ to a branch containing $T\,X$ and $T\,X \supset Y$. This is closely connected to *analytic cut*, allowing a cut rule that is restricted to subformulas of formulas already present. As a full system this originated in [23, 24], but was modified in [10] for FDE and investigated for its computational efficiency. This is something that might usefully be further explored in our present setting. We do not pursue it here.

Our tableau system for LP first appeared in [7], as did our system for FDE. Probably the main reason that the system for $\mathsf{K}_3$ did not appear there is that in the 1990's people were not commonly looking at consequence (which is interesting for $\mathsf{K}_3$) but only at validity (which is not, since $\mathsf{K}_3$ has no validities).

The tableau system for FDE in the well-known [26, Section 8.3] of Graham Priest has a close relationship with ours. Instead of prefixes, the use of $+$ and $-$ signs are combined with a special role for negation. Then what appears on a tableau branch is one of $X, +$ or $X, -$, where $X$ is a formula that may begin with a negation symbol, which is the ordinary symbol and not an 'outside' symbol like our signs. One can read $X, +$ as representing that $X$ is true and $X, -$ that it isn't.

Here is a translation from our signed version to the Priest version.

$$\begin{aligned} T\,X &\text{ becomes } X, + \\ \overline{T}\,X &\text{ becomes } X, - \\ F\,X &\text{ becomes } \neg X, + \\ \overline{F}\,X &\text{ becomes } \neg X, - \end{aligned}$$

This allows one to avoid introducing signs, but it is at the cost of overloading the use of the negation symbol. The overloading makes a translation in the other direction something of a problem. For instance, should $\neg X, -$ translate to $\overline{T}\,\neg X$ or to $\overline{F}\,X, -$? In [28] Smullyan faced exactly the same issue with his unsigned version of tableaus. We do not go into the issue further here. The tableau rules in [26] either correspond to ours directly or to rules that are admissible.

In a sense the present work has its ultimate origins in the book [20]. It is possible that this book had an influence on [7], though it is not cited as a reference. The book is cited in [25], which does things using sequent calculi instead of tableaus. Related semantical work can be found in [11].

## 7 Conclusion and Future Work

While there is an infinite family of prime logical Morgan lattices, the number of logics determined is very small. But the ones thus determined have played an important role, especially recently given the fundamental issues emerging from research on the strict/tolerant phenomena. Indeed $\mathsf{FDE}$, $\mathsf{K}_3$, and $\mathsf{LP}$ are much discussed currently. It a nice thing that all these can be formulated to have the same simple tableau rules as $\mathsf{CL}$. Or perhaps it should not be surprising since they are all sublogics of classical logic. There clearly is considerable unity to the family, and this is one of the ways it shows up. And, as we have seen, tableaus give a simple account of why the family is not bigger than it is.

There are two directions that look interesting for future research. One is intuitionistic logic, which can also be seen as a sublogic of classical logic but of a very different kind. Despite differences, connections are there, and hopefully will be the subject of a subsequent paper. The other direction is investigating what, if anything, can be said that is of interest concerning families of logics determined by weakening the lattice conditions of prime logical Morgan lattices. Here ideas stand on less certain ground, and it remains to be seen what emerges.

## References

[1] Alan Ross Anderson and Nuel D. Belnap Jr. *The Logic of Relevance and Necessity.* Vol. 1. Princeton University Press, 1975.

[2] Ofer Arielier Arieli and Arnon Avron. “Reasoning with logical bilattices”. In: *Journal of Logic, Language, and Information* 5.1 (1996), pp. 25–63.

[3] Florencio González Asenjo. “A calculus of antinomies”. In: *Notre Dame Journal of Formal Logic* 7.103-105 (1966).

[4] Nuel D. Belnap Jr. “A useful four-valued logic”. In: *Modern Uses of Multiple-Valued Logic.* Ed. by Jon Michael Dunn and George Epstein. Springer Netherlands, 1977, pp. 5–37.

[5] Nuel D. Belnap Jr. “A Useful Four-valued Logic: How a Computer Should Think”. In: *Entailment: The Logic of Relevance and Necessity.* Ed. by Nuel D. Belnap Jr. and J. M. Dunn. Vol. II. Princeton University Press, 1992, pp. 506–541.

[6] Kamila Bendová. "Interpolation and three-valued logics". In: *Reports on Mathematical Logic* 39 (2005), pp. 127–131.

[7] Anthony Bloesch. "A Tableau Style Proof System for Two Paraconsistent Logics". In: *Notre Dame Journal of Formal Logic* 34.2 (1993), pp. 295–301.

[8] Pablo Cobreros, Elio La Rosa, and Luca Tranchini. "(I Can't Get No) Antisatisfaction". In: *Synthese* 198 (2021), pp. 8251–8265.

[9] William Craig. "Linear reasoning. A new form of the Herbrand-Gentzen theorem." In: *Journal of Symbolic Logic* 22 (1957), pp. 250–268.

[10] Marcello D'Agostino. *Investigations into the complexity of some propositional calculi.* Tech. rep. Oxford University Computing Laboratory Programming Research Group, 1990.

[11] Thomas Macaulay Ferguson. "Notes on the Model Theory of DeMorgan Logics". In: *Notre Dame Journal of Formal Logic* 53.1 (2012), pp. 113–132.

[12] Hartry Field. *Saving Truth from Paradox.* Oxford University Press, 2008.

[13] Melvin Fitting. "A Family of Strict/Tolerant Logics". In: *Journal of Philosophical Logic* 50 (Apr. 2021). First published online, September 7, 2020., pp. 363–394.

[14] Melvin Fitting. "A Program to Compute Gödel-Löb Fixpoints". In: *Bulletin EATCS* 58 (1996), pp. 118–130.

[15] Melvin Fitting. *First-Order Logic and Automated Theorem Proving.* (First edition, 1990.) Errata at `http://melvinfitting.org/errata/errata.html`. Springer-Verlag, 1996.

[16] Melvin Fitting. *Intuitionistic Logic Model Theory and Forcing.* Amsterdam: North-Holland Publishing Co., 1969.

[17] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logics.* Dordrecht: D. Reidel Publishing Co., 1983.

[18] Melvin Fitting and Richard Mendelsohn. *First-Order Modal Logic.* (First Edition). Paperback, 1999. Errata at `http://melvinfitting.org/errata/errata.html`. Dordrecht: Kluwer, 1998.

[19] Melvin Fitting and Richard Mendelsohn. *First-Order Modal Logic.* Second Edition. See also [18]. Cham, Switzerland: Springer, 2023.

[20] Reiner Hähnle. *Automated Deduction in Multiple-Valued Logics.* Oxford University Press, 1993.

[21] Stephen C. Kleene. *Introduction to Metamathematics.* Princeton, NJ: D. Van Nostrand, 1950.

[22] Peter Milne. "A Non-classical refinement of the interpolation property for classical propositional logic". In: *Logique & Analyse* 235 (2016), pp. 273–281.

[23] Marco Mondadori. "Classical analytical deduction". In: *Annali dell'Universitá di Ferrara* Sez. III, Discussion paper series 1 (1988).

[24] Marco Mondadori. "Classical analytical deduction, part II". In: *Annali dell'Universitá di Ferrara* Sez. III, Discussion paper series 5 (1988).

[25] Multlog. *Analytic proof systems for Priest's logic of paradox LP.* URL: `https://logic.at/multlog/paradox.pdf` (visited on 2022).

[26] Graham Priest. *An Introduction to Non-Classical Logic: From If to Is.* Second Edition. Cambridge Introductions to Philosophy. First Edition published 2001. Cambridge University Press, 2008.

[27] Chris Scambler. "Classical logic and the strict tolerant hierarchy". In: *Journal of Philosophical Logic* 49 (2020), pp. 351–370.

[28] Raymond M. Smullyan. *First-Order Logic.* (Revised Edition, Dover Press, New York, 1994). Berlin: Springer-Verlag, 1968.

[29] Stefan Wintein and Reinhard Muskens. "Interpolation methods for Dunn logics and their extensions". In: *Studia Logica* 105 (2017). Special Issue: 40 Years of FDE, pp. 1319–1347.