TABLEAUX AND DECISION PROCEDURES FOR MANY-VALUED MODAL LOGICS

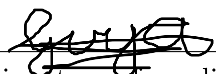
Guy Axelrod

A dissertation submitted to the Faculty of Science,
University of the Witwatersrand, Johannesburg,
in fulfilment of the requirements for the degree of Master of Science.

2024

DECLARATION

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



____

(Signature of candidate)

10th day of June 20 24 at 16:58

Acknowledgements

In short, I would like to thank my supervisor, Professor Conradie, for all his kindness and support. His guidance was invaluable.

I would also like to thank the DSI-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS) for providing funding.

Contents

List of Figures

Abstract

The aim of this dissertation is to present results expanding on the work done by Melvin Fitting in [22] and [24]. In [22], Fitting introduces a framework of many-valued modal logics, where modal formulas are interpreted via generalized Kripke models in which both the propositional valuation and the accessibility relation take on values from some Heyting algebra of truth values. For a fixed arbitrary finite Heyting algebra, $\mathcal{H}$, [24] presents a signed semantic tableau system that is sound and complete with respect to all $\mathcal{H}$-frames. We go on to consider the many-valued generalizations of frame properties such as reflexivity and transitivity (as presented in [39]) and give parameterized tableau systems which are sound and complete with respect to classes of $\mathcal{H}$-frames satisfying such properties. Further, a prefixed tableau system is introduced, which allows us to define an intuitive decision procedure deciding the logics of the above-mentioned $\mathcal{H}$-frame classes, as well as logics of $\mathcal{H}$-frames satisfying generalized symmetry properties, which cannot be captured by Fitting's unprefixed systems. Further, they allow us to derive finite frame properties. Such a decision procedure has been implemented, and is available on GitHub.

1 Introduction

Modal logics are non-classical logics that originated as an attempt to formalize modality. Modalities are ubiquitous in natural and technical languages, where they are words or phrases which can be applied to a statement to create a new statement that makes an assertion about where, when, how or in which situation the original statement is true. With the introduction of Kripke semantics as a general approach to interpreting modal languages, modal logic also became a powerful yet concise tool for describing relational structures. An extensive survey of this history can be found in [31].

In parallel, a separate family of non-classical logics, termed many-valued logic, has evolved from the pioneering works done by Lukasiewicz and Post in the 1920's (see [47]). And, since Zadeh's investigations on fuzzy sets in the 1970's, the field has garnered broader interest and popularity in a wide variety of applications. From areas in computer science (such as automated theorem proving, approximate reasoning, machine learning, multi-agent systems, program verification) to areas of pure mathematics (such as independence, generalized set theories) [32]. Many-valued logics allow a statement to take on one of many (possibly more than two) truth values. What the specific set of truth values is and how we interpret them usually depends on the application or theoretical interest. For example, Hájek popularized the use of real-valued (or 'fuzzy') logics to formalize reasoning with imprecise/vague notions [34].

In standard modal logic [1] both the worlds and the accessibility relation of the semantic structures (i.e., Kripke structures) are two-valued. The worlds are two-valued in the sense that there are only two values (0 and 1 or, *false* and *true*) that a formula can have at a world. The accessibility relation is two-valued (or, as is often said in the literature, 'crisp') in the sense that it can be modeled by a binary relation on worlds.

However, there is nothing inherently stopping one from going beyond two values into the many-valued setting. Indeed, this may be of great usefulness in applying logic to model the reasoning required for a particular problem, such as those where one is interested in a logical account of both modality and vagueness. As such, Many-valued modal logics have been applied to model and reason about problems in a wide range of settings. For example, [15] applies many-valued modal logic to the task of reasoning about fuzzy temporal relations. Many-valued generalizations of non-distributive modal logics have been

[1] By 'standard' is meant all those logics studied in standard reference texts in modal logic such as [5, 6, 12]

employed to model and reason about competition, based on their explanatory power, among scientific theories [14], and to capture certain phenomena of socio-political competition [17]. In [16] many-valued modal logics are enlisted into a framework for reasoning about vague-concepts and categorization.
There are numerous approaches to extending modal logic to a many-valued setting that have been previously studied. Some of the earliest are [49, 55, 41, 42, 43]. All of these early works focus on many-valued worlds and do not stray from crisp accessibility relations. In other words, the notion of a Kripke frame is not modified. The first framework to generalize modal logic with both many-valued worlds and many-valued accessibility relations (thus generalizing Kripke frames) arose in the early 1990's, with a series of papers by Melvin Fitting [21, 22]. Fitting provides an elegant justification for studying such logics as a way of capturing scenarios involving the opinions of multiple experts, where a dominance relation exists among the experts. Further, in [27], he explores the 'muddy children puzzle' as a possible use case for such logics. This dissertation is concerned with the particular approach to many-valued modal logic established in [22]. There, Fitting introduces $\mathcal{H}$-valued modal logics. More precisely, he defines an interpretation of modal formulas via generalized Kripke models, in which both propositions and accessibility relations take on values from an arbitrary finite Heyting algebra $\mathcal{H}$. Later in the 90's, Hajek et al. introduced similarly general fuzzy/real-valued modal logics [35, 36].

Broadly, what we shall do in this dissertation is explore certain proof theoretic aspects of Fitting's approach to many-valued modal logic. A study of the proof theory of these logics was initiated by Fitting himself when they were first introduced. Specifically, [22] gives a Gentzen sequent calculus for $\mathbf{K}^{\mathcal{H}}$ – the $\mathcal{H}$-valued analog of the basic modal logic $\mathbf{K}$. Koutras et al. [39] introduce $\mathcal{H}$-frame generalizations of standard Kripke frame properties such as seriality, reflexivity, symmetry and transitivity[2]. These generalized frame properties are parameterized by an arbitrary $\mathcal{H}$-value d, and for a given d, they define the logics $\mathbf{D}^{\mathcal{H}}_d, \mathbf{T}^{\mathcal{H}}_d, \mathbf{KB}^{\mathcal{H}}_d$ and $\mathbf{K4}^{\mathcal{H}}_d$ – the $\mathcal{H}$-valued analogs of the basic modal logics $\mathbf{D}, \mathbf{T}, \mathbf{KB}$ and $\mathbf{K4}$ respectively[3]. They then go on to extend Fitting's sequent calculus for $\mathbf{K}^{\mathcal{H}}$ to sequent calculi for these logics. This dissertation takes a similar path, however we focus on a different proof system than a Gentzen sequent system. Namely, we study tableau systems. The sequent calculi in [22] and [39] rely on a cut rule. In [24], Fitting defines a cut-free semantic tableau system for $\mathbf{K}^{\mathcal{H}}$. We shall go on to extend this system to cut-free tableau systems for $\mathbf{T}^{\mathcal{H}}_d, \mathbf{KB}^{\mathcal{H}}_d$ and $\mathbf{K4}^{\mathcal{H}}_d$, parameterized by some $\mathcal{H}$-value d.
$\mathbf{KB}^{\mathcal{H}}_d$ requires that we introduce a prefixed tableau system. And prefixed systems prompts us to define a decision procedure for the satisfiability and validity problems for these logics.

Let us now briefly survey the proof theoretic advances made for many-valued modal logics. In [44], Priest introduces tableau systems (as well as nice philosophical applications) for certain four and three-valued crisp modal logics. His tableau system is essentially a prefixed one, which, along with the prefixed systems defined in [20], provide the underlying inspiration for the prefixed system presented in this work. In [8, 9], a broad basis for the study of many-valued modal logics based on finite residuated lattices is established, thus generalizing Fitting's work. Since then, there has been much work on the axiomatizibility and decidability of various many-valued modal logics. Vidal has contributed much to this area, and good overviews and references can be found in [56, 57]. To the best of our knowledge, much of this recent work shifts focus from Fittings finite valued Heyting semantics to more fuzzy, real valued semantics. The works most similar to this one are [53, 19, 48], in that they focus on Fitting's framework. [53] provides a cut-free sequent calculus for $\mathbf{K}^{\mathcal{H}}$, and as such, is essentially the first work to provide a decidability

[2]These generalizations were also established in [28]

[3]The value d acts in specifying the degree to which a certain frame property holds. For instance, to say that a many-valued frame is d-reflexive implies that each world sees itself to degree at least d.

result for this logic. [19] and [48] study tableaux for the crisp versions of the logics we consider here. In particular, [19] provides prefixed tableau systems for such crisp logics with very general modalities. It is not entirely clear however how to adapt that work to the non-crisp setting, and Section 4 may be viewed as a step in that direction. Also very worth noting is the possibility of translating the logics we deal with to appropriate first order many-valued logics. Questions regarding decision procedures for these logics are studied by Hähnle in [33], where he describes a kind of prefixed tablea system.

The dissertation is organized as follows. In Section 2 we formally present the concepts, as well as elementary results regarding these concepts, that will be of interest. In Section 3, we study unprefixed tableau systems. We start by describing Fitting's system for $\mathbf{K}^{\mathcal{H}}$, along with relevant results from [24]. We then go on to introduce unprefixed tableau systems for the logics $\mathbf{T}^{\mathcal{H}}_d$ and $\mathbf{K4}^{\mathcal{H}}_d$, as well as proofs for soundness, strong completeness and compactness results. Completeness in this section is proven using a 'Henkin-style' argument. Having considered these unprefixed systems, we are in a position to contrast them to the prefixed systems that we proceed to introduce.
Section 4 starts by defining a prefixed tableau system for $\mathbf{K}^{\mathcal{H}}$. It is then shown that the system is sound and weakly complete with respect to the class of all $\mathcal{H}$-frames. By way of proving completeness, we describe a decision procedure for $\mathbf{K}^{\mathcal{H}}$. We end the section by briefly considering the changes needed to define a prefixed tableau system and decision procedure for $\mathbf{KB}^{\mathcal{H}}_d$.

2 Preliminaries

In this section, we formally present all the relevant background material. As mentioned in the introduction, we will consider a family of many-valued modal logics first introduced by Melvin Fitting in [22]. This involves replacing the classically two-valued boolean algebra truth space with an arbitrary finite Heyting algebra. Therefore, we embark on introducing Heyting Algebras immediately in Section 2.1. After this, we go on to Section 2.2, in which we formally define the syntax and semantics of the logics we will be concerned with. Then, in Section 2.3, we provide an abstract presentation of the tableau systems that will be studied, along with the asscociated general terminology and results which will later be used. Finally, Section 2.4 describes labeled trees - structures that will be used extensively when defining our tableau-based decision procedures.

2.1 Heyting Algebras

For a detailed exposition of Heyting algebras and related topics, see [45]. This subsection treats that material which is immediately relevant to this dissertation. One may approach defining Heyting Algebras either in terms of orderings or purely algebraically (that is to say, as abstract algebraic structures satisfying certain equational axioms). We choose the order theoretic approach, as it lends itself well to visual intuitions (via Hasse diagrams). Whereas, the abstraction provided by the axiomatic approach does not provide any clear advantage in the current context. As such, let us begin by introducing the basic order theoretic notions required.

Definition 2.1. A ***partially ordered set*** is a tuple $(H, \leq)$ where H is a non-empty set and $\leq$ is a binary relation on H satisfying the following conditions for all $a, b, c \in H$:

1. $a \leq a$,
2. If $a \leq b$ and $b \leq c$, then $a \leq c$,

3. If $a \leq b$ and $b \leq a$, then $a = b$.

Conditions (1), (2) and (3) are respectively called the reflexivity, transitivity and antisymmetry of $\leq$. Instead of $a \leq b$, we may also write $b \geq a$. We then say that b is above (or greater than) a, and a is below (or less than) b.
We may also make use of the shorthand $a < b$ to indicate that $a \leq b$ and $a \neq b$.

Example 2.2. Consider an arbitrary set A.

1. Denote the power set of A (that is, the set of all subsets of A) by $\mathcal{P}(A)$. The tuple $(\mathcal{P}(A), \subseteq)$ is a partially ordered set, where $\subseteq$ is the usual subset relation on sets.

2. The natural numbers along with the usual 'less than or equal to' relation is a partially ordered set.

3. Suppose A is non-empty. Then $(A, =)$ is a partially ordered set.

It is often more natural to present and think of partially ordered sets diagrammatically. This is facilitated by the following definition.

Definition 2.3. Let $(H, \leq)$ be a partially ordered set. For $a, b \in H$, we say a is ***covered*** by b (or b covers a) iff $a < b$ and there does not exist an element $c \in H$ for which $a < c < b$.

Now, a partially ordered set $(H, \leq)$ is uniquely determined by a diagram constructed as follows: Represent each element of H as a vertex in the plane, and draw an edge going upwards from vertex a to vertex b iff a is covered by b.
Such a diagram is called a ***Hasse diagram*** for $(H, \leq)$.

Example 2.4. Below is a Hasse Diagram for the partially ordered set $(\mathcal{P}(\{1,2\}), \subseteq)$. Note that the order relation (in this case $\subseteq$) can be inferred by considering the upward paths in the diagram.

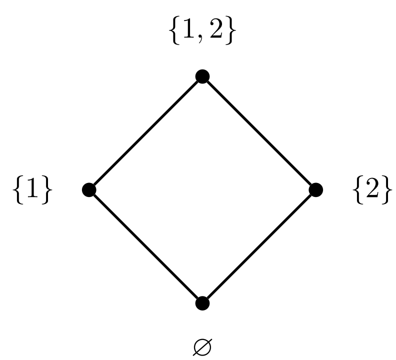


Figure 2.1

Definition 2.5. Let $(H, \leq)$ be a partially ordered set and $G \subseteq H$. An element $a \in G$ is said to be a ***greatest*** (***least***) element of G iff $b \leq a$ ($a \leq b$) for all $b \in G$.

Remark 2.6. If a, a' are greatest (least) elements of G, then, by definition, $a \leq a'$ and $a' \leq a$. So, $a = a'$ by the antisymmetry of $\leq$. From this we can see that there is at most one greatest (least) element of G.

Definition 2.7. Let $(H, \leq)$ be a partially ordered set. For $G \subseteq H$, we say $a \in G$ is a ***maximal (minimal)*** element of G iff for all $b \in G$, $a \leq b$ $(b \leq a)$ implies $a = b$.
We shall use $max(G)$ $(min(G))$ to denote the set of all maximal (minimal) elements of G.
There are various equivalent forms of the above definition that may be more appealing in certain contexts. For example, $a \in G$ is a maximal (minimal) element of G iff there is no element $b \in G$ such that $a < b$ $(b < a)$.

Remark 2.8. Using an induction argument on the size of the subset, it is straightforward to show that $max(G)$ and $min(G)$ are non-empty for every non-empty *finite* subset G.

Lemma 2.9. *Let $(H, \leq)$ be a partially ordered set. For every finite subset $G \subseteq H$, if $b \in G$ then,*

1. *$b \leq a$ for some $a \in max(G)$*
2. *$a \leq b$ for some $a \in min(G)$*

Proof. The argument for (2) is the dual of that for (1). As such, we only present the argument for (1).
We shall prove by induction that for all $k \in \mathbb{N}$,

$$\text{for every } G \subseteq H \text{ with } |G| = k, \text{ if } b \in G \text{ then } b \leq a \text{ for some } a \in max(G). \tag{2.9.1}$$

The base case requires us to show that (2.9.1) holds for $k = 0$ and $k = 1$.
For $k = 0$, (2.9.1) is vacuously true.
For $k = 1$, let $G \subseteq H$ with $|G| = k$. Then $G = \{a\}$ for some $a \in H$. So if $b \in G$, then $b = a$ and clearly $a \in max(G)$. That is, $b \leq a$ for some $a \in max(G)$.

Now let $k \geq 1$ and, as our induction hypothesis, assume (2.9.1) holds for k. We wish to show that (2.9.1) holds for $k + 1$.
Let $G \subseteq H$ with $|G| = k + 1$ and suppose $b \in G$. By Remark 2.8, there exists some $a \in max(G)$. We consider two cases:

<u>Case 1</u> $b \leq a$. Then the desired result holds and we are done.

<u>Case 2</u> $b \not\leq a$. Then $b \neq a$, and so $b \in G \backslash \{a\}$. Hence, by the induction hypothesis, $b \leq a'$ for some $a' \in max(G \backslash \{a\})$. And since $b \not\leq a$, we must have $a' \not\leq a$. Thus, $a' \in max(G)$.

This concludes the induction proof. Now, if we consider any finite $G \subseteq H$, we have $|G| = k$ for some $k \in \mathbb{N}$. So (1) reduces to (2.9.1). □

Remark 2.10. The above Lemma is usually presented in a more general form. Instead of assuming that G is finite, one assumes that every chain in G has an upper bound in G. Then proceed via transfinite induction (see [45, p. 33] for details). However, such generality will not be necessary for this dissertation, as we will only be concerned with finite partially ordered sets.

Lemma 2.11. *Let $(H, \leq)$ be a finite partially ordered set. For all $a, b \in H$,*

1. *$a \leq b$ iff $b \not\leq u$ for all $u \in max(\{c \in H \mid a \not\leq c\})$.*
2. *$b \leq a$ iff $u \not\leq b$ for all $u \in min(\{c \in H \mid c \not\leq a\})$.*

Proof. Let $a, b \in H$. The argument for (2) is the dual of that for (1). As such, we only present the argument for (1).

- For the forward implication, suppose $a \leq b$ and let $u \in max(\{c \in H \mid a \nleq c\})$. If $b \leq u$, then by transitivity, $a \leq u$. Therefore, since $u \in \{c \in H \mid a \nleq c\}$, we must have $b \nleq u$.

- For the converse implication, we prove the contrapositive.
  Suppose $a \nleq b$. Then $b \in \{c \in H \mid a \nleq c\}$. But $\{c \in H \mid a \nleq c\}$ is finite since H is finite. Therefore, by Lemma 2.9, there must exist some $u \in max(\{c \in H \mid a \nleq c\})$ such that $b \leq u$.

□

Definition 2.12. Let $(H, \leq)$ be a partially ordered set and $G \subseteq H$. An eleement $a \in H$ is said to be an ***upper (lower) bound*** of G iff $b \leq a$ $(a \leq b)$ for all $b \in G$.
If the set of all upper (lower) bounds of G contains a least (greatest) element, then Remark 2.6 tells us that this element is unique. We call it the ***supremum (infimum)*** of G and denote it by $\sup G$ $(\inf G)$.

Note that the supremum (infimum) does not necessarily exist for a subset of a partially ordered set. For instance, consider Example 2.2 (3) and take a subset of A with at least two elements.
The case in which every two-element subset of a partially ordered set has a supremum and infimum is of particular interest to us.

Definition 2.13. A partially ordered set $(H, \leq)$ is a ***lattice*** iff every two-element subset of H has a supremum and infimum. In this case we define the $\wedge$ (meet) and $\vee$ (join) operations as follows. For all $a, b \in H$,

$$a \wedge b := \inf\{a, b\},$$
$$a \vee b := \sup\{a, b\}.$$

If there exists a least and greatest element of H, then the lattice is said to be ***bounded***. The least and greatest elements of a bounded lattice are unique and will be denoted by 0 and 1 respectively. Henceforth, we will call 0 the ***bottom*** element, and 1 the ***top*** element.
A lattice is said to be ***distributive*** iff for all $a, b, c \in H$,

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \qquad \text{and} \qquad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

The next Proposition collects various properties of lattices that follow easily from the definitions. They will be used, often without comment, throughout the text.

Proposition 2.14. *Let $(H, \leq)$ be an arbitrary lattice. For every $a, b, c, d \in H$,*

$$a \vee a = a, \qquad a \wedge a = a, \tag{2.14.1}$$
$$a \leq a \vee b, \qquad a \wedge b \leq a, \tag{2.14.2}$$
$$a \leq c, b \leq c \text{ implies } a \vee b \leq c, \qquad c \leq a, c \leq b \text{ implies } c \leq a \wedge b, \tag{2.14.3}$$
$$a \leq b \text{ iff } a \vee b = b, \qquad b \leq a \text{ iff } a \wedge b = b, \tag{2.14.4}$$
$$a \leq c, b \leq d \text{ implies } a \vee b \leq c \vee d, \qquad a \leq c, b \leq d \text{ implies } a \wedge b \leq c \wedge d, \tag{2.14.5}$$
$$a \vee b = b \vee a, \qquad a \wedge b = b \wedge a, \tag{2.14.6}$$
$$a \vee (b \vee c) = (a \vee b) \vee c, \qquad a \wedge (b \wedge c) = (a \wedge b) \wedge c, \tag{2.14.7}$$
$$(a \wedge b) \vee b = b, \qquad a \wedge (a \vee b) = a. \tag{2.14.8}$$

2.14.1, 2.14.6, 2.14.7 and 2.14.8 are called the idempotent, commutative, associative and absorption laws respectively. Taken together, they can be used as the equational axioms defining the class of lattices. That is to say, for any abstract algebraic structure satisfying these identities, we can define an ordering such that we get a lattice in which the algebraic operations correspond to the meet and join operations in the lattice.

Associativity lets us ignore brackets when dealing with finite meets and joins. And, by a simple induction argument, it can be shown that for a finite $G = \{a_1, \ldots, a_n\} \subseteq H$, we have

$$a_1 \wedge \ldots \wedge a_n = \inf G,$$
$$a_1 \vee \ldots \vee a_n = \sup G.$$

This prompts us to generalize meets and joins to arbitrary, possibly infinite sets as follows.

Definition 2.15. Let $(H, \leq)$ be a lattice. For any subset $G \subseteq H$, finite or infinite, we define

$$\bigwedge G \coloneqq \inf G,$$
$$\bigvee G \coloneqq \sup G.$$

Remark 2.16. For a bounded lattice, it follows from the definitions that $\bigwedge \varnothing = 1$ and $\bigvee \varnothing = 0$.

The following results will be used in later proofs.

Proposition 2.17. *Let $(H, \leq)$ be an arbitrary lattice and $\{a_i\}_{i\in I}$ an arbitrary subset of H, where I is an index set. For every $a \in H$, if the relevant joins and meets exist, then*

$$\bigvee \{a_i\}_{i\in I} \leq a \text{ iff } a_i \leq a \text{ for all } i \in I, \tag{2.17.1}$$
$$a \leq \bigwedge \{a_i\}_{i\in I} \text{ iff } a \leq a_i \text{ for all } i \in I, \tag{2.17.2}$$
$$\text{if } a_i \leq b_i \text{ for every } i \in I, \text{ then } \bigvee \{a_i\}_{i\in I} \leq \bigvee \{b_i\}_{i\in I}, \tag{2.17.3}$$
$$\text{if } a_i \leq b_i \text{ for every } i \in I, \text{ then } \bigwedge \{a_i\}_{i\in I} \leq \bigwedge \{b_i\}_{i\in I}, \tag{2.17.4}$$
$$\bigvee \{a \vee a_i\}_{i\in I} = a \vee \bigvee \{a_i\}_{i\in I}, \tag{2.17.5}$$
$$\bigwedge \{a \wedge a_i\}_{i\in I} = a \wedge \bigwedge \{a_i\}_{i\in I}. \tag{2.17.6}$$

Definition 2.18. Let $(H, \leq)$ be a bounded lattice and $a, c \in H$. We call c a ***complement of*** a iff

$$a \wedge c = 0 \quad \text{and} \quad a \vee c = 1$$

Definition 2.19. A partially ordered set $(H, \leq)$ is said to be a ***Boolean algebra*** iff it is a bounded distributive lattice and every element $a \in H$ has a complement. In this case, the complement is uniquely determined by a, and we denote it by $\neg a$.

Example 2.20. The simplest Boolean algebra is the two-valued Boolean algebra

$$\mathbf{2} = (\{0, 1\}, \{(0, 0), (0, 1), (1, 1)\})$$

.

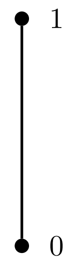


Figure 2.2

Boolean algebras play a central role in the algebraic semantics of two-valued logics, including standard modal logic. For details, see [5, Chapter 5].
We now consider an approach to generalizing the idea of complements in a lattice.

Definition 2.21. Let $(H, \leq)$ be a lattice and $a, b, c \in H$. We call c the ***pseudo-complement of** a* ***relative to*** b iff c is the greatest element of $\{c' \in H \mid a \wedge c' \leq b\}$, or equivalently, for every $d \in H$,

$$d \leq c \text{ iff } a \wedge d \leq b.$$

If such a c exists, then it is unique, and we denote it by $a \Rightarrow b$.

The element $a \Rightarrow b$ does not necessarily exist in an arbitrary lattice. This leads us to the main notion of this subsection.

Definition 2.22. A partially ordered set $(H, \leq)$ is said to be a ***Heyting algebra*** iff it is a bounded lattice in which the element $a \Rightarrow b \in H$ exists for every $a, b \in H$.

Example 2.23. Below we give Hasse diagrams determining partially ordered sets that are Heyting algebras.

1. The three-valued Heyting algebra, which we will often denote by $\mathcal{H}^3 = (\{0, h, 1\}, \leq)$.

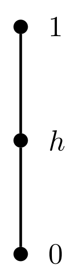


Figure 2.3

2. A Heyting algebra $\mathcal{H}^5$ with five elements.

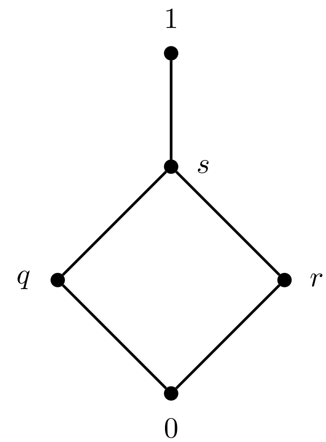


Figure 2.4

In a Boolean algebra, we may retrieve the relative pseudo-complement operation through the complement by defining $a \Rightarrow b \coloneqq \neg a \vee b$.
On the other hand, if we are given a Heyting algebra, we may define the unary *pseudo-complement* operation $\neg$ by $\neg a \coloneqq (a \Rightarrow 0)$. Then, it is easy to show that $a \wedge \neg a = 0$, but in general it is not necessarily the case that $a \vee \neg a = 1$. If the Heyting algebra is a Boolean algebra, then the pseudo-complement coincides with the complement.
Analogous to the connection between Boolean algebras and classical propositional logic, Heyting algebras model the algebraic structure of intuitionistic logic (see [4]).

The following Proposition collects results about Heyting algebras that will be useful in later proofs.

Proposition 2.24. *Let $(H, \leq)$ be an arbitrary Heyting algebra. For every $a, b, c, d \in H$,*

$$a \leq (b \Rightarrow c) \text{ iff } (a \wedge b) \leq c \text{ iff } b \leq (a \Rightarrow c), \tag{2.24.1}$$
$$a \Rightarrow b = 1 \text{ iff } a \leq b, \tag{2.24.2}$$
$$1 \Rightarrow a = a, \tag{2.24.3}$$
$$a \leq b \text{ implies } b \Rightarrow c \leq a \Rightarrow c, \tag{2.24.4}$$
$$a \leq b \text{ implies } c \Rightarrow a \leq c \Rightarrow b, \tag{2.24.5}$$
$$b \leq a \Rightarrow b, \tag{2.24.6}$$
$$a \wedge (a \Rightarrow b) = a \wedge b \leq b, \tag{2.24.7}$$
$$(a \Rightarrow b) \wedge b = b, \tag{2.24.8}$$
$$a \Rightarrow (b \wedge c) = (a \Rightarrow b) \wedge (a \Rightarrow c), \tag{2.24.9}$$
$$(a \vee b) \Rightarrow c = (a \Rightarrow c) \wedge (b \Rightarrow c), \tag{2.24.10}$$
$$a \Rightarrow (b \Rightarrow c) = (a \wedge b) \Rightarrow c = b \Rightarrow (a \Rightarrow c). \tag{2.24.11}$$

Further, for $\{a_i\}_{i\in I} \subseteq H$, if the meets and joins in question exist, then

$$a \vee \bigwedge\{a_i\}_{i\in I} \leq \bigwedge\{a \vee a_i\}_{i\in I}, \tag{2.24.12}$$

$$a \wedge \bigvee\{a_i\}_{i\in I} = \bigvee\{a \wedge a_i\}_{i\in I}, \tag{2.24.13}$$

$$\bigvee\{a_i \Rightarrow a\}_{i\in I} \leq \bigwedge\{a_i\}_{i\in I} \Rightarrow a, \tag{2.24.14}$$

$$\bigvee\{a \Rightarrow a_i\}_{i\in I} \leq a \Rightarrow \bigvee\{a_i\}_{i\in I}, \tag{2.24.15}$$

$$\bigwedge\{a_i \Rightarrow a\}_{i\in I} = \bigvee\{a_i\}_{i\in I} \Rightarrow a, \tag{2.24.16}$$

$$\bigwedge\{a \Rightarrow a_i\}_{i\in I} = a \Rightarrow \bigwedge\{a_i\}_{i\in I}. \tag{2.24.17}$$

The next result, which may seem quite arbitrary, will be used later. However, for the time being, it illustrates some of the ways that we will use the properties presented in the previous proposition.

Lemma 2.25. *Let $(H, \leq)$ be an arbitrary Heyting algebra. For any $a', b', c', d, d', e' \in H$,*

1. $d \wedge \big(a' \Rightarrow (b' \wedge (d \Rightarrow c'))\big) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big) \leq a' \Rightarrow (d' \wedge (d \Rightarrow e'))$
2. $d \wedge \big((b' \vee (d \wedge c')) \Rightarrow a'\big) \wedge \big((d' \vee (d \wedge e')) \Rightarrow c'\big) \leq (d' \vee (d \wedge e')) \Rightarrow a'$

Proof. 1. Distributing $\Rightarrow$ over $\wedge$, we have that

$$\big(a' \Rightarrow (b' \wedge (d \Rightarrow c'))\big) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big) = (a' \Rightarrow b') \wedge (a' \Rightarrow (d \Rightarrow c')) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big)$$

So it suffices to show that

$$d \wedge (a' \Rightarrow b') \wedge (a' \Rightarrow (d \Rightarrow c')) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big) \leq a' \Rightarrow (d' \wedge (d \Rightarrow e'))$$

To do so, we take an arbitrary $a \in H$ and argue that if

$$a \leq d \wedge (a' \Rightarrow b') \wedge (a' \Rightarrow (d \Rightarrow c')) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big)$$

then, $a \leq a' \Rightarrow (d' \wedge (d \Rightarrow e'))$.
Let $a \in H$ and suppose $a \leq d \wedge (a' \Rightarrow b') \wedge (a' \Rightarrow (d \Rightarrow c')) \wedge \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big)$. Then $a \leq d$, $a \leq (a' \Rightarrow (d \Rightarrow c'))$ and $a \leq \big(c' \Rightarrow (d' \wedge (d \Rightarrow e'))\big)$. Or, equivalently,

$$a \wedge d = a, \tag{2.25.1}$$

$$a \wedge d \wedge a' \leq c', \tag{2.25.2}$$

$$a \wedge c' \leq d' \wedge (d \Rightarrow e'). \tag{2.25.3}$$

Since $a \wedge d \wedge a' \leq a$, with (2.25.2) we have $a \wedge d \wedge a' \leq a \wedge c'$. So, by (2.25.3) and the transitivity of $\leq$, we may conclude that $a \wedge d \wedge a' \leq d' \wedge (d \Rightarrow e')$. I.e., $a \wedge d \leq a' \Rightarrow (d' \wedge (d \Rightarrow e'))$. So by (2.25.1) it follows that $a \leq a' \Rightarrow (d' \wedge (d \Rightarrow e'))$, as required.

2. By the properties of $\Rightarrow$ and $\vee$, we have

$$\big((b' \vee (d \wedge c')) \Rightarrow a'\big) \wedge \big((d' \vee (c \wedge e')) \Rightarrow c'\big) = (b' \Rightarrow a') \wedge ((d \wedge c') \Rightarrow a') \wedge \big((d' \vee (d \wedge e')) \Rightarrow c'\big)$$

So it suffices to show that

$$d \wedge (b' \Rightarrow a') \wedge ((d \wedge c') \Rightarrow a') \wedge \big((d' \vee (d \wedge e')) \Rightarrow c'\big) \leq (d' \vee (d \wedge e')) \Rightarrow a'$$

To do so, we show that for every $a \in H$, if

$$a \leq d \wedge (b' \Rightarrow a') \wedge ((d \wedge c') \Rightarrow a') \wedge \big((d' \vee (d \wedge e')) \Rightarrow c'\big)$$

then $a \leq (d' \vee (d \wedge e')) \Rightarrow a'$.
Let $a \in H$ and suppose $a \leq d \wedge (b' \Rightarrow a') \wedge ((d \wedge c') \Rightarrow a') \wedge \big((d' \vee (d \wedge e')) \Rightarrow c'\big)$. Then $a \leq d$, $a \leq ((d \wedge c') \Rightarrow a')$ and $a \leq \big((d' \vee (d \wedge e')) \Rightarrow c'\big)$. The last two conditions are equivalent to

$$a \wedge d \wedge c' \leq a', \tag{2.25.4}$$

$$a \wedge (d' \vee (d \wedge e')) \leq c'. \tag{2.25.5}$$

Since $a \wedge (d' \vee (d \wedge e')) \leq a$, with (2.25.5), we have that $a \wedge (d' \vee (d \wedge e')) \leq a \wedge d \wedge c'$. So, by (2.25.4) and the transitivity of $\leq$, we may conclude that $a \wedge (d' \vee (d \wedge e')) \leq a'$. I.e., $a \leq (d' \vee (d \wedge e')) \Rightarrow a'$, as required.

□

As is the case with lattices, we can define the class of Boolean algebras and the class of Heyting algebras in a purely algebraic fashions. For details see [45]. Suffice it to say that the variety of Boolean algebras is properly contained in the variety of Heyting algebras, which is in turn contained in the variety of distributive lattices. Further, it can be shown that any finite distributive lattice is in fact a Heyting algebra. In the rest of this dissertation, we will be dealing with finite Heyting algebras. As such, they can be equivalently thought of as finite distributive lattices.

2.2 Many-Valued Modal Logic

Finite Heyting algebras will act as the truth value spaces of our logics. The syntax and semantics of the many valued logics we study depend on the specific Heyting algebra we choose to act as the truth value space. So, let us once and for all fix an arbitrary finite Heyting algebra $\mathcal{H} = (H, \leq)$. We continue to use $\wedge, \vee, \Rightarrow, 0, 1$ for the meet, join, relative pseudo-complement, bottom and top respectively. Henceforth, we make no further assumptions about $\mathcal{H}$ (except possibly in examples), and as such, all that follows effectively constitutes a study of a family of many-valued modal logics indexed by a particular Heyting algebra.

2.2.1 Syntax

Let us refer to elements of H as ***$\mathcal{H}$-truth values***. In general, not every $\mathcal{H}$-truth value is definable in terms of the Heyting algebra operations. So, we explicitly add them to the language of our logic. Let us fix some finite set of propositional constants $\underline{H}$, where each symbol $\underline{a} \in \underline{H}$ represents a specific $\mathcal{H}$-truth value a. Let us also fix some non-empty countable [4] set Φ of propositional variables, whose elements we will denote by p or p_i for $i \in \mathbb{N}$.
The language for our many-valued modal logic, which we denote by $\mathcal{L}^{\mathcal{H}}(\Phi)$, consists of finite strings constructed from the alphabet $\underline{H} \cup \Phi \cup \{\wedge, \vee, \supset, \Diamond, \Box, (,)\}$. $\mathcal{L}^{\mathcal{H}}(\Phi)$ itself is countable.
The particular strings of this language that we will deal with are the ***formulas***. The set of $\mathcal{H}$-valued modal formulas (or simply 'formulas' from now on), denoted $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$, can be defined inductively as follows:

[4] Countability is required for Lemma 2.68. Further, it ensures that each propositional variable has a finite encoding, which is required for any decision procedure taking formulas as inputs.

Definition 2.26. $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is the smallest set satisfying

1. $\Phi \cup \underline{H} \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$.
2. If $\varphi, \psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$, then the following are in $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$:
   - $(\varphi \wedge \psi)$
   - $(\varphi \vee \psi)$
   - $(\varphi \supset \psi)$
   - $\Diamond\varphi$
   - $\Box\varphi$

Remark 2.27. Where there is no danger of confusion, we will usually drop the outermost brackets of a formula.

The base case formulas i.e., the strings in $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ by virtue of condition (1) of the above definition, are called the ***atomic*** formulas.

Remark 2.28. We use $\equiv$ to express the syntactic identity between formulas.

The inductive nature of $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ allows us to prove that a certain property holds for all formulas, by making use of induction on the structure of formulas. Indeed, we shall frequently make use of this strategy. Further, $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is freely generated. That is, given a formula, it has a unique decomposition into exactly one of the following forms:

$$\underline{a},\ p,\ (\varphi \wedge \psi),\ (\varphi \vee \psi),\ (\varphi \supset \psi),\ \Diamond\varphi,\ \Box\varphi,$$

where $\underline{a} \in H$, $p \in \Phi$ and $\varphi, \psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. This allows us to uniquely define functions and relations on formulas via recursion. Most importantly, the semantics will be defined in this fashion. But first, some syntactic definitions and functions regarding formulas which will be useful.

Definition 2.29. $Sub : Frm(\mathcal{L}^{\mathcal{H}}(\Phi)) \to \mathcal{P}(Frm(\mathcal{L}^{\mathcal{H}}(\Phi)))$,
$Mdegree : Frm(\mathcal{L}^{\mathcal{H}}(\Phi)) \to \mathbb{N}$ are the unique functions satisfying the following for all $\varphi, \psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ and $\gamma \in \Phi \cup \underline{H}$:

- $Sub(\gamma) = \{\gamma\}$, $degree(\gamma) = 0$, $Mdegree(\gamma) = 0$.
- For $\otimes \in \{\wedge, \vee, \supset\}$, $Sub((\varphi \otimes \psi)) = \{(\varphi \otimes \psi)\} \cup Sub(\varphi) \cup Sub(\psi)$, $degree((\varphi \otimes \psi)) = 1 + degree(\varphi) + degree(\psi)$, $Mdegree((\varphi \otimes \psi)) = Mdegree(\varphi) + Mdegree(\psi)$.
- For $\odot \in \{\Diamond, \Box\}$, $Sub(\odot\varphi) = \{\odot\varphi\} \cup Sub(\varphi)$, $degree(\odot\varphi) = 1 + degree(\varphi)$, $Mdegree(\odot\varphi) = 1 + Mdegree(\varphi)$.

We call an element of $Sub(\varphi)$ a ***subformula*** of φ.
$degree(\varphi)$ indicated the number of connectives in φ. $Mdegree(\varphi)$ is the ***modal degree*** of φ, and indicates the number of occurrences of the symbols $\Diamond, \Box$ in φ.

Remark 2.30. It is routine to show, by induction on the *degree*, that for each formula φ, we have $|Sub(\varphi)| \leq 2 \times degree(\varphi) + 1$

Definition 2.31. A formula $\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is a ***bounding implication*** iff $\varphi \equiv \underline{a} \supset \psi$ or $\varphi \equiv \psi \supset \underline{a}$ for some $\underline{a} \in \underline{H}$ and $\psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$.

Definition 2.32. Let φ be a formula. Define a ***bounded subformula*** of φ to be any bounding implication of the form $\underline{a} \supset \psi$ or $\psi \supset \underline{a}$, where $\underline{a} \in \underline{H}$ and $\psi \in Sub(\varphi)$.

Remark 2.33. For any formula φ, the set of all bounded subformulas of φ has at most $2 \times |\underline{H}| \times |Sub(\varphi)|$ elements. Hence, since H is finite, there is a finite number of bounded subformulas of φ.

Negation, in the standard sense, is not available to us. As a result, we introduce new symbols which have 'negation-like' semantics (see Definition 2.48) useful for defining tableau rules.

Definition 2.34. Let T and F be two new formal symbols. The set of ***signed formulas*** is

$$SFrm(\mathcal{L}^{\mathcal{H}}(\Phi)) := \{T\varphi \mid \varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))\} \cup \{F\varphi \mid \varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))\}.$$

In other words, a signed formula is just a formula with either the symbol T or F prepended to it.

Remark 2.35. Let $s\varphi$ be a signed formula, where $s \in \{T, F\}$. We can extend the definition of *degree* and *Mdegree* by simply ignoring the sign:

- $degree(s\varphi) = degree(\varphi)$.
- $Mdegree(s\varphi) = Mdegree(\varphi)$.

2.2.2 Semantics

Now, let us consider the semantic structures in which formulas from $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ are interpreted. These structures are many-valued generalizations of classical Kripke structures, and were introduced in [23, 21, 22].

Definition 2.36. An $\mathcal{H}$***-frame*** is a tuple $\mathfrak{F} = (W, R)$, where W is a non-empty set of ***worlds*** and $R : W \times W \to H$ is a function assigning $\mathcal{H}$-truth values to pairs of worlds.

The function R can be thought of as an $\mathcal{H}$-valued accessibility relation. Or, if we think of classical Kripke frames as directed graphs where the accessibility relation defines the edges, then we may think of $\mathcal{H}$-frames as complete directed graphs in which edges are weighted by R.

Definition 2.37. An $\mathcal{H}$***-model*** for $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is a structure $\mathfrak{M} = ((W, R), V)$, where $\mathfrak{F} = (W, R)$ is an $\mathcal{H}$-frame (we say that $\mathfrak{M}$ is based on frame $\mathfrak{F}$) and V is a ***valuation*** on $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. By this, we mean that $V : W \times (\Phi \cup \underline{H}) \to H$ is a function assigning $\mathcal{H}$-truth values to atomic formulas in each world, such that

$$V(\mathfrak{s}, \underline{a}) = a$$

for all $\mathfrak{s} \in W$ and $\underline{a} \in H$. That is, propositional constants are always mapped by a valuation to the $\mathcal{H}$-truth values that they represent.

Remark 2.38. As mentioned in the introduction, there have been many other approaches to many-valued modal semantic structures different to those just presented. Indeed, Fitting himself originally also considered two other somewhat simpler semantics: one where only propositions are many-valued (the crisp case) and one where only the accessibility relation is many-valued. Recently, it has been shown how crisp many-valued modal logics (based on algebras even more general than Heyting algebras) can be polynomially embedded within standard modal logic [2].

Remark 2.39. Fitting suggests that $\mathcal{H}$-models provide natural models of the epistemic stances of committees of experts also taking into account the influence relations among them. In this light, possible applications of the logics we study may include the problem of "mixing experts" in the context of machine learning [60].

Given an ***$\mathcal{H}$-model***, we can extend the valuation to all formulas in $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ via a recursive definition.

Definition 2.40. Let $\mathfrak{M} = ((W, R), V)$ be an $\mathcal{H}$-model. The extension of V, $\overline{V} : W \times Frm(\mathcal{L}^{\mathcal{H}}(\Phi)) \to H$, is the unique function where for any $\mathfrak{s} \in W$ and $\varphi, \psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$, we have

- $\overline{V}(\mathfrak{s}, \gamma) = V(\mathfrak{s}, \gamma)$ for every $\gamma \in \Phi \cup \underline{H}$,
- $\overline{V}(\mathfrak{s}, (\varphi \wedge \psi)) = \overline{V}(\mathfrak{s}, \varphi) \wedge \overline{V}(\mathfrak{s}, \psi)$,
- $\overline{V}(\mathfrak{s}, (\varphi \vee \psi)) = \overline{V}(\mathfrak{s}, \varphi) \vee \overline{V}(\mathfrak{s}, \psi)$,
- $\overline{V}(\mathfrak{s}, (\varphi \supset \psi)) = \overline{V}(\mathfrak{s}, \varphi) \Rightarrow \overline{V}(\mathfrak{s}, \psi)$,
- $\overline{V}(\mathfrak{s}, \Box\varphi) = \bigwedge\{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \overline{V}(\mathfrak{v}, \varphi) \mid \mathfrak{v} \in W\}$,
- $\overline{V}(\mathfrak{s}, \Diamond\varphi) = \bigvee\{R(\mathfrak{s}, \mathfrak{v}) \wedge \overline{V}(\mathfrak{v}, \varphi) \mid \mathfrak{v} \in W\}$.

Henceforth, we employ the harmless abuse of notation in which V is used to denote both a valuation and its extension.
We say that φ is ***satisfied*** by $\mathfrak{M} = ((W, R), V)$ at $\mathfrak{s} \in W$ (denoted as $\mathfrak{M}, \mathfrak{s} \Vdash \varphi$) iff $V(\mathfrak{s}, \varphi) = 1$.
Further, φ is ***globally satisfied*** by $\mathfrak{M}$ (denoted as $\mathfrak{M} \Vdash \varphi$) iff $V(\mathfrak{s}, \varphi) = 1$ for every $\mathfrak{s} \in W$. We say $\mathfrak{M}$ is a ***counter model*** for φ iff $\mathfrak{M} \nVdash \varphi$.

Since we are dealing with a finite Heyting algebra $\mathcal{H}$, all the meets and joins in the above definition exist.
In cases in which no semantic ambiguity arises, brackets in a formula will often be removed.

Remark 2.41. In line with Fitting's presentation, we use $\wedge, \vee$ to denote the meet and join operations in $\mathcal{H}$ as well as symbols occurring in $\mathcal{L}^{\mathcal{H}}(\Phi)$. Context should make it clear exactly which objects we are referring to. Further, the use of an underline for elements of $\underline{H}$ will help differentiate between syntactic and semantic objects. For example, for the $\mathcal{H}$-truth value $a \wedge b$ where $a, b \in H$, one should understand $\underline{a \wedge b}$ as referring to the constant symbol in our language that maps to the semantic object resulting from the evaluation of the meet operation.

Recall that pseudo-complementation in Heyting algebras is defined by $\neg a := (a \Rightarrow 0)$. In the standard two-valued case, complementation in Boolean algebras reflects the logical notion of negation. Inspired by this, we will introduce the following abbreviation for formulas $\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$:

$$\neg\varphi \equiv (\varphi \supset \underline{0}).$$

It should be noted that if $\mathcal{H}$ is the Boolean algebra **2** consisting of two elements, then the many-valued model logic we have introduced reduces to the standard two-valued modal logic. In this standard case, it is clear that some of our connectives are redundant, as for instance: $\Box\varphi$ is semantically equivalent to $\neg\Diamond\neg\varphi$; and $\varphi \vee \psi$ is semantically equivalent to $\neg(\neg\varphi \wedge \neg\psi)$. However, in the general case, the connectives we have in our language are not interdefinable in this way (see Example 2.42). As such, we need to explicitly include them.

Earlier, we considered $\mathcal{H}$-frames as weighted, complete directed graphs. Now, we present a graphical representation of an $\mathcal{H}$-model and calculate the $\mathcal{H}$-truth value of a specific formula at some world. But first, we make a simple but useful observation. With the properties of lattices and Heyting algebras in mind when reflecting on the modal cases in Definition 2.40, it is evident that the edges weighted with 0 do not play a role in determining the value of a formula. Therefore, we lose nothing by excluding these edges from the graph.

Example 2.42. This example is taken from page 7 of [22]. We let $\mathcal{H} = \mathcal{H}^3$. The below graph represents an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$, where $W = \{\mathfrak{s}, \mathfrak{v}\}$, $R = \{((\mathfrak{s}, \mathfrak{s}), 0), ((\mathfrak{s}, \mathfrak{v}), 1), ((\mathfrak{v}, \mathfrak{s}), 0), ((\mathfrak{v}, \mathfrak{v}), 0)\}$, $V(\mathfrak{s}, p) = 0$, $V(\mathfrak{v}, p) = h$.

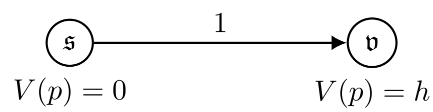


Figure 2.5

Consider the formula $\varphi \equiv \neg\Diamond\neg p \supset \Box p$. We calculate the value of φ at a state in a model by recursively unraveling the valuation until we bottom out at the values for the atomic formulas, at which point we evaluate the resulting algebraic expression. For instance, at $\mathfrak{s}$ in $\mathfrak{M}$, we have

$$\begin{aligned}
V(\mathfrak{s}, \Box p) &= \bigwedge\{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow V(\mathfrak{v}, p) \mid \mathfrak{v} \in W\} \\
&= (R(\mathfrak{s}, \mathfrak{s}) \Rightarrow V(\mathfrak{s}, p)) \wedge (R(\mathfrak{s}, \mathfrak{v}) \Rightarrow V(\mathfrak{v}, p)) \\
&= (0 \Rightarrow 0) \wedge (1 \Rightarrow h) \\
&= 1 \wedge h \\
&= h.
\end{aligned}$$

And,

$$\begin{aligned}
V(\mathfrak{s}, \Diamond\neg p) &= V(\mathfrak{s}, \Diamond(p \supset \underline{0})) \\
&= \bigvee\{R(\mathfrak{s}, \mathfrak{v}) \wedge V(\mathfrak{v}, p \supset \underline{0}) \mid \mathfrak{v} \in W\} \\
&= \big(R(\mathfrak{s}, \mathfrak{s}) \wedge V(\mathfrak{s}, p \supset \underline{0})\big) \vee \big(R(\mathfrak{s}, \mathfrak{v}) \wedge V(\mathfrak{v}, p \supset \underline{0})\big) \\
&= \big(R(\mathfrak{s}, \mathfrak{s}) \wedge (V(\mathfrak{s}, p) \Rightarrow V(\mathfrak{s}, \underline{0}))\big) \vee \big(R(\mathfrak{s}, \mathfrak{v}) \wedge (V(\mathfrak{v}, p) \Rightarrow V(\mathfrak{v}, \underline{0}))\big) \\
&= \big(0 \wedge (0 \Rightarrow 0)\big) \vee \big(1 \wedge (h \Rightarrow 0)\big) \\
&= 0 \vee 0 \\
&= 0.
\end{aligned}$$

So, $V(\mathfrak{s}, \neg\Diamond\neg p) = V(\mathfrak{s}, \Diamond\neg p \supset \underline{0}) = V(\mathfrak{s}, \Diamond\neg p) \Rightarrow 0 = 0 \Rightarrow 0 = 1$. Hence,

$$\begin{aligned}
V(\mathfrak{s}, \varphi) &= V(\mathfrak{s}, \neg\Diamond\neg p \supset \Box p) \\
&= V(\mathfrak{s}, \neg\Diamond\neg p) \Rightarrow V(\mathfrak{s}, \Box p) \\
&= 1 \Rightarrow h \\
&= h \neq 1.
\end{aligned}$$

We now define the main semantic notions that will be studied.

Definition 2.43 (**Validity**). Let $\mathfrak{F} = (W, R)$ be an $\mathcal{H}$-frame and $\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. We say that φ is ***valid*** in $\mathfrak{F}$ (denoted as $\mathfrak{F} \Vdash \varphi$) iff for every $\mathcal{H}$-model $\mathfrak{M} = (\mathfrak{F}, V)$ based on $\mathfrak{F}$, we have $\mathfrak{M} \Vdash \varphi$ (or equivalently, there does not exist a counter model, based on $\mathfrak{F}$, for φ).
We will mainly be concerned with classes of $\mathcal{H}$-frames. Let $\mathcal{F}$ be some class of $\mathcal{H}$-frames. φ is said to be valid in $\mathcal{F}$, or $\mathcal{F}$-valid (denoted as $\mathcal{F} \Vdash \varphi$) iff $\mathfrak{F} \Vdash \varphi$ for all $\mathfrak{F} \in \mathcal{F}$. In the case where $\mathcal{F}$ is the class of all $\mathcal{H}$-frames, we simply say that φ is valid.
We define $\Lambda_{\mathcal{F}}$ to be $\{\varphi \in Frm(\mathcal{L}^{\mathcal{H}}) \mid \mathcal{F} \Vdash \varphi\}$, and call it the ***logic of*** $\mathcal{F}$.

Definition 2.44 (**Entailment**). Let $\mathcal{F}$ be some class of $\mathcal{H}$-frames, and $\Gamma \cup \{\varphi\} \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. We say that Γ $\mathcal{F}$-***entails*** φ iff for every $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$ and $\mathfrak{s} \in W$ such that $(W, R) \in \mathcal{F}$, if $\mathfrak{M}, \mathfrak{s} \Vdash \gamma$ for each $\gamma \in \Gamma$, then $\mathfrak{M}, \mathfrak{s} \Vdash \varphi$. In the case where $\mathcal{F}$ is the class of all $\mathcal{H}$-frames, we simply say that Γ entails φ.

Remark 2.45. Note that validity is a special case of entailment. Namely, $\mathcal{F} \Vdash \varphi$ iff φ is $\mathcal{F}$-entailed by the empty set.

Historically, in the context of standard modal logic, various classes of frames have been characterized in terms conditions on the two-valued accessibility relation and extensively studied. We will be concerned with classes of $\mathcal{H}$-frames which are characterized by many-valued generalizations of some of these conditions, as defined in [39].

Definition 2.46. Let $\mathfrak{F} = (W, R)$ be an $\mathcal{H}$-frame and d an $\mathcal{H}$-truth value.

1. The ***theory/logic of*** all $\mathcal{H}$-frames is the set of formulas
$$\mathbf{K}^{\mathcal{H}} := \{\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi)) \mid \varphi \text{ is valid in every } \mathcal{H}\text{-frame}\}.$$

2. We say that $\mathfrak{F}$ is d-**reflexive** iff
$$R(\mathfrak{s}, \mathfrak{s}) \geq d$$
for every $\mathfrak{s} \in W$.
Let $\mathsf{Refl}_d^{\mathcal{H}}$ denote the class of all d-reflexive $\mathcal{H}$-frames.
The ***theory/logic of*** $\mathsf{Refl}_d^{\mathcal{H}}$ is the set of formulas
$$\mathbf{T}_d^{\mathcal{H}} := \Lambda_{\mathsf{Refl}_d^{\mathcal{H}}}.$$

3. We say that $\mathfrak{F}$ is d-**transitive** iff
$$d \wedge R(\mathfrak{s}, \mathfrak{v}) \wedge R(\mathfrak{v}, \mathfrak{r}) \leq R(\mathfrak{s}, \mathfrak{r})$$
for all $\mathfrak{s}, \mathfrak{v}, \mathfrak{r} \in W$.
Let $\mathsf{Trans}_d^{\mathcal{H}}$ denote the class of all d-transitive $\mathcal{H}$-frames.
The ***theory/logic of*** $\mathsf{Trans}_d^{\mathcal{H}}$ is the set of formulas
$$\mathbf{K4}_d^{\mathcal{H}} := \Lambda_{\mathsf{Trans}_d^{\mathcal{H}}}.$$

4. We say that $\mathfrak{F}$ is d-**serial** iff

$$\bigvee\{R(\mathfrak{s},\mathfrak{v}) \mid \mathfrak{v} \in W\} \geq d$$

for every $\mathfrak{s} \in W$.
Let $\mathsf{Ser}_d^{\mathcal{H}}$ denote the class of all d-serial $\mathcal{H}$-frames.
The ***theory/logic of*** $\mathsf{Ser}_d^{\mathcal{H}}$ is the set of formulas

$$\mathbf{D}_d^{\mathcal{H}} := \Lambda_{\mathsf{Ser}_d^{\mathcal{H}}}.$$

5. We say that $\mathfrak{F}$ is d-**symmetric** iff

$$d \wedge R(\mathfrak{s},\mathfrak{v}) = d \wedge R(\mathfrak{v},\mathfrak{s})$$

for every $\mathfrak{s},\mathfrak{v} \in W$.
Let $\mathsf{Symm}_d^{\mathcal{H}}$ denote the class of all d-symetric $\mathcal{H}$-frames.
The ***theory/logic of*** $\mathsf{Symm}_d^{\mathcal{H}}$ is the set of formulas

$$\mathbf{KB}_d^{\mathcal{H}} := \Lambda_{\mathsf{Symm}_d^{\mathcal{H}}}.$$

Remark 2.47. The names of the theories are in keeping with historical convention. The definitions collapse to the standard case when $d = 1$ and $\mathcal{H} = \mathbf{2}$. For instance, $\mathbf{T}_1^{\mathbf{2}}$ is the same as the standard modal logic $\mathbf{T}$ of reflexive Kripke frames.
The names in standard modal logic derive from the names for the axioms defining the frame properties. We are further justified in using these names since when we take these axioms to the $\mathcal{H}$-valued setting, the generalized frame properties we gave above are still defined by them. [10] gives a good account of why this is so.

Definition 2.48. Given some $\mathcal{H}$-model $\mathfrak{M} = ((W,R),V)$ and $\mathfrak{s} \in W$, we shall say that a signed formula is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$ iff it is $T\varphi$ and $V(\mathfrak{s},\varphi) = 1$; or it is $F\varphi$ and $V(\mathfrak{s},\varphi) \neq 1$.
Let $\mathcal{F}$ be a class of $\mathcal{H}$-frames and X a set of signed formulas. X is ***$\mathcal{F}$-satisfiable*** iff there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W,R),V)$ and a world $\mathfrak{s} \in W$ such that $(W,R) \in \mathcal{F}$ and each member of X is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In the case where $\mathcal{F}$ is the class of all $\mathcal{H}$ frames, we simply say that X is satisfiable.

2.3 Tableau Systems

To verify that a formula is valid in some class of frames via a direct appeal to the semantic definitions requires us to step out of the formal system we are studying and argue (in a higher system which we take to be semiformal mathematicas, and call the ***metasystem***) about infinite objects in a generally non-constructive manner.
On the other hand, a *formal proof*, in the most general sense, is a syntactic object (that is to say, consisting entirely of strings of formal symbols) which acts as a finitary certificate of validity for a formula.
A ***proof system*** aims to provide a framework for defining and constructing formal proofs. Constructing a formal proof involves a finite sequence of *rule* applications, which simply modify, add and/or delete the strings in a syntactic object[5]. Axiomatic, or 'Hilbert-style', systems, along with being the oldest, are perhaps the most conceptually clear and elegant systems. They lend themselves well to proving various

[5] In some cases, reasonable proof systems do not even exist. More formally speaking, there are logics that are not *recursively axiomatizable*. For many-valued modal examples, see [57].

metatheorems about the logic we are studying. However, axiomatic proofs are generally hard to discover. Other proof systems may not be as conceptually clear, but lend themselves better to the actual task of finding a formal proof, in that they will constrain our search in some way. Such systems are good candidates for automation. For classical logic, *resolution* systems are the basis of highly efficient automated theorem provers [25]. One of the reasons resolution systems are so effective is because they are an example of 'backward reasoning' systems. In such a system, constructing a formal proof for φ involves starting with φ in some form, and deconstructing it in some manner until we end up with atomic objects.
We shall be studying another type of backward reasoning formal systems, called *semantic tableau* (from now on simply tableau) systems. These systems strike a good balance between theoretical usefulness and automatability. Tableau systems were first introduced by Beth [3] and Popularized by Smullyan [50]. They have since been widely adapted to be used for various non-classical logics [18]. Fitting gives a detailed account of its use for standard modal logics in [20], and this particular text motivated much of the work in this dissertation. Another influential text is [46], in which tableau systems are employed towards proving interpolation for modal logics.

In the context of tableau systems, a formal proof is a syntactic object called a (closed) tableau (plural tableaux). Let us call the set of strings that can occur in a formal proof an *object language*. Before precisely defining tableaux, we need to define the relevant object languages.

Definition 2.49. A ***signed bounding implication*** is a signed formula in which the formula is a bounding implication. The set of all signed bounding implication is contained in $SFrm(\mathcal{L}^{\mathcal{H}}(\Phi))$, and will be denoted as SBI.
We shall use $\perp$ as an abbreviation for the signed bounding implication $F(\underline{0} \supset \underline{1})$.

Definition 2.50. Let $\beta \in SBI$, $\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ and $a \in H$. We say that β ***bounds*** φ ***by*** a iff β is of the form $T(\underline{a} \supset \varphi), T(\varphi \supset \underline{a}), F(\underline{a} \supset \varphi)$ or $F(\varphi \supset \underline{a})$.

Definition 2.51. We say $\beta \in SBI$ is ***atomic*** iff β bounds φ by some $a \in H$ where φ is an atomic formula.

SBI will play the role of object language in what we call unprefixed tableaux. The object language for what will be called prefixed tableaux involve, unsurprisingly, prefixes – which we define now.

Definition 2.52. Fix some countably infinite set of symbols Σ. A ***prefix*** is a tuple (w, σ), where $w \in \Sigma$ and $\sigma \subseteq \Sigma \times \Sigma \times \underline{H}$.

For an intuition of how prefixes should be interpreted, see Section 4. However, for the time being it suffices to just observe that they consist of abstract formal symbols.

Definition 2.53. A ***prefixed signed bounding implication*** is a string of the form $(w, \sigma)\beta$, consisting of a prefix (w, σ) prepended to a signed bounding implication β.
We denote the set of all prefixed signed bounding implication by $pSBI$.

Remark 2.54. In the strictest sense, prefixes are not strings and cannot form part of other strings. Rather, they are tuples in which the second item is a relation, and this makes them easier to talk about in Section 4. However, it is not much of a stretch to devise a way of encoding prefixes as strings, and when it makes sense to do so (as in the above definition) we can implicitly assume that prefixes are strings.

We expand on the Tableau System defined by Fitting in [24]. Fitting presents his tableaux in the tradition of Smullyan [50], in which a tableau is a tree where each node is labelled by a single signed bounding implication.
More abstractly, we can fully describe all the necessary details of such an object through the use of a collection in $\mathcal{P}(\mathcal{P}(SBI))$ (i.e., a set of sets of signed bounding implications). Such an abstract approach has the advantage of unifying the notions of tableau used throughout this dissertation (unprefixed and prefixed) and avoids some of the vagueness that surrounds the destructive/forgetful nature of modal rule application in [24].

Let U be our object language (so $U = SBI$ or $U = pSBI$). The set of tableaux for some formula will be defined recursively as a subset of $\mathcal{P}(\mathcal{P}(U))$ that results from applying a finite sequence of permissible operations on some base tableau. The permissible operations are described via what we call ***tableau rules***.
A tableau rule $\rho = (\mathcal{N}, (\mathcal{D}_1, \dots, \mathcal{D}_n), \textit{side condition})$ consists of a *numerator* $\mathcal{N}$, a finite list of *denominators* $\mathcal{D}_1, \dots, \mathcal{D}_n$, and a *side condition*. Schematically, ρ is presented as follows.

$$(\rho)\ \begin{array}{c|c|c} \multicolumn{3}{c}{\mathcal{N}} \\ \hline \mathcal{D}_1 & \dots & \mathcal{D}_n \end{array} \qquad\qquad \textit{side condition}$$

The numerator, denominators and side condition of a tableau rule are expressions of the metalanguage that are used to describe subsets of U based on the membership of certain elements adhering to a particular syntactic form and syntactic conditions stated in the *side condition*. An ***instantiation*** of the numerator and denominator(s) of a rule are the sets that can result from a uniform substitution of sets, constants and formulas for metasymbols in the rule, such that the *side condition* is satisfied. Further, the numerators and denominators of the rules we will consider denote one or more distinguished elements of U. We will call (the instantiation) of such an element the ***principal element*** of the numerator/denominator. This is best illustrated by example.

Example 2.55. Suppose we are working with $U = SBI$ and the Heyting algebra $\mathcal{H}^5$ as presented in Example 2.23. Consider the following tableau rule Ex. It is a very contrived and convoluted rule, but it illustrates all the possible types of syntactic operations that the unprefixed tableau rules to follow may require.

$$(\text{Ex})\ \begin{array}{l|c|l} \multicolumn{3}{c}{X; F(\underline{a} \supset (\underline{b} \vee (\varphi \wedge \psi)))} \\ \hline X; & \dots & X; \\ T(\varphi \supset (\underline{b \wedge t_1} \vee \psi)) & & T(\varphi \supset (\underline{b \wedge t_n} \vee \psi)) \end{array}$$

Where $t_1, \dots, t_n$ are all the maximal $\mathcal{H}$-truth values not above a, and $a \neq 0$.

We use the following metasymbols in a rule: X for an element of $\mathcal{P}(U)$; Greek letters φ, ψ, θ for elements of $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$; lower case roman letters such as a, b, t_i, u_i for $\mathcal{H}$-truth values, and $\underline{a}, \underline{b}, \underline{t_i}, \underline{u_i}$ for the elements of $\underline{H}$ that represent the truth values. Further, for $X \in \mathcal{P}(U)$ and $x \in U$, we use $X; x$ as a shorthand for $X \cup \{x\}$.
Then, suppose we uniformly substitute X with $\{T(\underline{q} \supset (p_1 \supset p_2), T(p_1 \supset \underline{1}))\}$; a with s; b with r; φ with $(p_1 \vee p_2)$; and ψ with $\underline{q}$. Then the set

$$\{T(\underline{q} \supset (p_1 \supset p_2)), T(p_1 \supset \underline{1}), F(\underline{s} \supset (\underline{r} \vee ((p_1 \vee p_2) \wedge \underline{q})))\}$$

is an instantiation of the numerator, with principal element $F(\underline{s} \supset (\underline{r} \vee ((p_1 \vee p_2) \wedge \underline{q})))$.

$$\{T(\underline{q} \supset (p_1 \supset p_2), T(p_1 \supset \underline{1})), T((p_1 \vee p_2) \supset (\underline{r} \vee \underline{q}))\}$$

and

$$\{T(\underline{q} \supset (p_1 \supset p_2), T(p_1 \supset \underline{1})), T((p_1 \vee p_2) \supset (\underline{0} \vee \underline{q}))\}$$

are the corresponding instantiations of the denominators, with principal elements $T((p_1 \vee p_2) \supset (\underline{r} \vee \underline{q}))$ and $T((p_1 \vee p_2) \supset (\underline{0} \vee \underline{q}))$ respectively. Note that in this rule, the number of denominators depends on what we substitute for a.

Remark 2.56. At first glance, it may appear that some tableau rules do not only involve trivial symbol checking and manipulation, they also require calculations in the context of the underlying Heyting algebra (as is the case in the previous example, in which we are concerned with finding maximal elements of a specific subset as well as evaluations of the meet operation). However, since we are dealing with finite Heyting algebras only, we can assume there exist a finite encoding for our algebra, along with effective symbolic procedures that can answer the algebraic questions that our tableau rules may ask.

As mentioned, the purpose of a tableau rule $\rho = (\mathcal{N}, (\mathcal{D}_1, \ldots, \mathcal{D}_n), \textit{side condition})$ is to describe a family of operations that can be applied to elements of $\mathcal{P}(\mathcal{P}(U))$. To be more precise, let $f : \mathcal{P}(\mathcal{P}(U)) \to \mathcal{P}(\mathcal{P}(U))$. We say f is *described* by ρ iff for all $T \in \mathcal{P}(\mathcal{P}(U))$, if $T \neq f(T)$ then for some $S \in T$, S is an instantiation of $\mathcal{N}$, $f(T)$ contains $S_1, \ldots S_n$ which are corresponding instantiations of $\mathcal{D}_1, \ldots, \mathcal{D}_n$ respectively, and $T \setminus \{S\} = f(T) \setminus \{S_1, \ldots, S_n\}$.[6]
In most cases we will not make explicit reference to an operation described by a rule. If $T^* = f(T)$ for some $T \in \mathcal{P}(\mathcal{P}(U))$ and f described by ρ, we shall say that T^* was derived from T through an ***application*** of ρ.[7] Sometimes, it will be useful to pick out the element of T which acts as the instantiation of the numerator of the rule. So, if $S \in T$ but $S \notin T^*$, we may say ρ was applied to S to derive T^*. We may still have cause to go even further and pick out the principal element x of S. In that case we will simply say that ρ was applied to S ***around*** principal element x.

A tableau system is a finite collection of tableau rules. Let us fix some tableau system $\mathcal{C} = \{\rho_1, \ldots, \rho_n\}$.

Definition 2.57. Let X be a finite subset of U. The set of ***$\mathcal{C}$-tableaux for*** X is a subset of $\mathcal{P}(\mathcal{P}(U))$ and is defined recursively as follows.

- $\{X\}$ is a $\mathcal{C}$-tableau for X
- Suppose T is a $\mathcal{C}$-tableau for X. If $T^* \in \mathcal{P}(\mathcal{P}(U))$ can be derived from T by applying some $\rho \in \mathcal{C}$, then T^* is a $\mathcal{C}$-tableau for X.

Further, the set of all $\mathcal{C}$-tableaux is simply the set of all $T \in \mathcal{P}(\mathcal{P}(U))$ such that T is a $\mathcal{C}$-tableau for some finite $X \subseteq U$.
We call the sets in a $\mathcal{C}$-tableau its ***branches***.

Remark 2.58. Note that we only define tableaux for finite subsets of U. This is in keeping with the spirit of formal proofs being finitary objects. However, this is not entirely necessary from a technical standpoint, as the nature of the rules we will deal with ensure that the notion of derivability is compact even if we allowed infinite subsets of U.

[6]Note that the identity operation on $\mathcal{P}(\mathcal{P}(U))$ is described by every rule.
[7]We may speak of an identity application of ρ in the case where f is the identitiy operation

Remark 2.59. An equivalent formulation of the proceeding definition is 'bottom-up'. More precisely, we have that T is a $\mathcal{C}$-tableau for X iff there exists a finite sequence $(T_1, \ldots, T_n)$ of terms in $\mathcal{P}(\mathcal{P}(U))$, such that $T_1 = \{X\}$, $T = T_n$, and for each $1 \leq i \leq n-1$, T_{i+1} is derived from T_i via an application of some $\rho_i \in \mathcal{C}$.

If $U = pSBI$, we say that $\mathcal{C}$ is a ***prefixed*** tableau system, and that $\mathcal{C}$-tableaux are prefixed tableaux. Otherwise, if $U = SBI$, we say that $\mathcal{C}$ is an ***unprefixed*** tableau system and that $\mathcal{C}$-tableaux are unprefixed tableaux. For the time being, let use just consider the case of $U = SBI$, until we deal with prefixed tableaux in Section 4.

Definition 2.60 (derivability). Given some set $S \in \mathcal{P}(U)$, we shall say that S is ***closed*** iff $\bot \in S$. Otherwise, we say that S is ***open***.
A tableau is closed iff all its branches are closed; otherwise it is open.
Consider a set of formulas $\Gamma \cup \{\varphi\}$. We say that φ is ***$\mathcal{C}$-derivable*** from Γ (denoted $\Gamma \vdash_{\mathcal{C}} \varphi$) iff for some finite subset $\Gamma_0 \subseteq \Gamma$, there exists a closed $\mathcal{C}$-tableau for $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma_0\} \cup \{F(\underline{1} \supset \varphi)\}$.
A formula φ is a ***theorem*** of $\mathcal{C}$ iff $\varnothing \vdash_{\mathcal{C}} \varphi$ (i.e., there exists a closed $\mathcal{C}$-tableau T for $\{F(\underline{1} \supset \varphi)\}$). In this case we also say that φ is provable in $\mathcal{C}$ (denoted as $\vdash_{\mathcal{C}} \varphi$), or that T is a $\mathcal{C}$-proof of φ.

We wish to establish a correspondence between the syntactic notion of theoremhood and the semantic notion of validity. We do so by establishing soundness and completeness.

Definition 2.61. Let $\mathcal{C}$ be some tableau system and $\mathcal{F}$ a class of $\mathcal{H}$-frames. We say that $\mathcal{C}$ is ***sound*** with respect to $\mathcal{F}$ iff for every formula φ, if $\vdash_{\mathcal{C}} \varphi$ then φ is $\mathcal{F}$-valid.
Conversely, we say that $\mathcal{C}$ is (weakly) ***complete*** with respect to $\mathcal{F}$ iff for every formula φ, if φ is $\mathcal{F}$-valid then $\vdash_{\mathcal{C}} \varphi$.
We will be concerned with tableau systems that are sound and complete with respect to the frame classes in Definition 2.46. Suppose $\mathcal{F}$ is such a frame class and Λ the theory of $\mathcal{F}$. Call $\mathcal{C}$ a tableau system ***for*** Λ iff the set of theorems of $\mathcal{C}$ is the same as Λ, or equivalently, $\mathcal{C}$ is sound and complete with respect to the $\mathcal{F}$. We will use '$\mathcal{C}\mathbf{X}$' to name the tableau system for Λ, where $\mathbf{X}$ is the name for Λ as given in Definition 2.46.

As is usual in logic, soundness is easier to show than completeness. To prove a tableau system $\mathcal{C}$ is sound with respect to a class of $\mathcal{H}$-frames $\mathcal{F}$ we essentially have to show that rule applications preserve $\mathcal{F}$-satisfiability.

Definition 2.62. Let $\mathcal{F}$ be a class of $\mathcal{H}$-frames. A $\mathcal{C}$-tableau T is ***$\mathcal{F}$-satisfiable*** iff at least one branch $S \in T$ is $\mathcal{F}$-satisfiable.
Consider some rule $\rho \in \mathcal{C}$. We say ρ ***preserves $\mathcal{F}$-satisfiability*** iff for every $\mathcal{C}$-tableau T, if T is $\mathcal{F}$-satisfiable and T^* is a tableau that can be derived from T via an application of ρ, then T^* is $\mathcal{F}$-satisfiable.

Lemma 2.63. *Let $\mathcal{F}$ be a class of $\mathcal{H}$-frames, and $\mathcal{C}$ a tableau system. If each rule of $\mathcal{C}$ preserves $\mathcal{F}$-satisfiability and there exists a closed $\mathcal{C}$-tableau for a finite set $X \in \mathcal{P}(U)$, then X is not $\mathcal{F}$-satisfiable.*

Proof. Suppose there exists a closed $\mathcal{C}$-tableau T^* for X. Assume X is $\mathcal{F}$-satisfiable; we derive a contradiction.
By Definition 2.57, any construction of a closed $\mathcal{C}$-tableau for X begins with the single branch tableau $T = \{X\}$. T is $\mathcal{F}$-satisfiable since, by assumption, X is $\mathcal{F}$-satisfiable. So, using the fact that each rule preserves $\mathcal{F}$-satisfiability in a simple induction argument on the number of rule applications, we can show that every subsequent $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-tableau we construct, including T^*, must be $\mathcal{F}$-satisfiable. But T^* is closed, and no closed tableau is $\mathcal{F}$-satisfiable. □

Proposition 2.64 (**Soundness**). *Let $\mathcal{F}$ be a class of $\mathcal{H}$-frames, and $\mathcal{C}$ a tableau system. If each rule of $\mathcal{C}$ preserves $\mathcal{F}$-satisfiability, then $\mathcal{C}$ is sound with respect to $\mathcal{F}$.*

Proof. Consider an arbitrary formula φ and suppose it is a theorem of $\mathcal{C}$. I.e. there exists a closed $\mathcal{C}$-tableau for $\{F(\underline{1} \supset \varphi)\}$. Therefore, by Lemma 2.63, $\{F(\underline{1} \supset \varphi)\}$ is not $\mathcal{F}$-satisfiable. That is, for every $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$, where $(W, R) \in \mathcal{F}$, we have $V(\mathfrak{s}, \underline{1} \supset \varphi) = 1$ for all $\mathfrak{s} \in W$. Which is equivalent to saying that $\mathfrak{M}, s \Vdash 1$ for all $\mathfrak{s} \in W$. Thus, φ is valid in $\mathcal{F}$. $\square$

Proving completeness requires more work. For the unprefixed systems, we will prove a stronger result.

Definition 2.65. Let $\mathcal{C}$ be some tableau system and $\mathcal{F}$ a class of $\mathcal{H}$-frames. We say that $\mathcal{C}$ is ***strongly complete*** with respect to $\mathcal{F}$ iff for every set of formulas $\Gamma \cup \{\varphi\}$, if Γ $\mathcal{F}$-entails φ, then $\Gamma \vdash_{\mathcal{C}} \varphi$.

Remark 2.66. Clearly weak completeness is the special case of strong completeness for $\Gamma = \varnothing$.
We will often simply speak of completeness, in which case it should be assumed we are talking about the weak version.

To show (strong) completeness, it is customary to prove the contrapositive: If φ is not $\mathcal{C}$-derivable from Γ, then φ is not $\mathcal{F}$-entailed by Γ. One common approach (and the one we take for the unprefixed systems) is to use a 'Henkin-style' argument in which an infinite canonical model, which satisfies $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma\} \cup \{F(\underline{1} \supset \varphi)\}$, is constructed from maximal-consistent sets.

Definition 2.67. A set $X \in \mathcal{P}(U)$ is said to be ***$\mathcal{C}$-consistent*** iff for all finite $X' \subseteq X$ there does not exist a closed $\mathcal{C}$-tableau for X'.
Further, X is ***maximal*** $\mathcal{C}$-consistent iff it is $\mathcal{C}$-consistent and any proper superset of X is not $\mathcal{C}$-consistent.

The following result concerning the existence of maximal consistent sets will be important in our 'Henkin-style' argument.

Proposition 2.68. *For any $\mathcal{C}$-consistent set X, there exists some maximal $\mathcal{C}$-consistent set X' for which $X \subseteq X'$.*

Proof. We say a collection of sets A is of *finite character* provided $X \in A$ iff every finite subset of X is in A.
The following result appears in [54]:

> If A is of finite character and $\bigcup A$ is countable, then for every $X \in A$, X is a subset of some element of A which is maximal with respect to the subset relation.

The proof employs a generalized Lindenbaum construction (see [51, p. 60] or [50, p. 38]).
Now, let A be the collection of all $\mathcal{C}$-consistent sets and apply this result to A. It follows from the definition of consistency that A is of finite character. Further, $\bigcup A \subseteq SFrm(\mathcal{L}^{\mathcal{H}}(\Phi))$, and $SFrm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is countable. $\square$

Remark 2.69. The general result in the above proof is closely related to *Tukey's Lemma*, which can be shown to be equivalent to the axiom of choice (See [37]).

2.4 Labeled Trees

Labeled trees will be a useful data structure when we come to Section 4. Here we define exactly what we mean by a labeled tree, and collect the basic terminology and results that will be used.

Definition 2.70. A ***directed graph*** is a tuple $\mathscr{G} = (N, E)$, where N is a non-empty set of elements called ***nodes***, and E (the which we call the accessibility relation) is a binary relation on N. For $n, n' \in N$, we say that n is accessible from n' iff $(n', n) \in E$.
A ***path*** $\mathscr{P}$ in $\mathscr{G}$ is any finite or countable sequence of nodes such that each term of the sequence, except the first, is accessible from the previous term.
We can identify a path with a string of names for the nodes in the path. Then, by a ***subpath*** of $\mathscr{P}$ we simply mean a substring of $\mathscr{P}$.

Definition 2.71. A ***tree*** is a directed graph $\mathscr{T} = (N, E)$ satisfying the following conditions:

1. There is a unique node $n \in N$ such that there is no node $n' \in N$ for which $(n', n) \in E$. This unique node we call the ***root*** of $\mathscr{T}$,
2. For each nodes $n \in N$ other than the root, there exists a unique node $n' \in N$ such that $(n', n) \in E$.
3. For any node $n \in N$, there exists a path in $\mathscr{T}$ starting at the root and ending at n.

Suppose $(n', n) \in E$. In the context of trees, instead of saying that n is accessible from n', we shall say that n' is a ***parent*** of n, and n is a ***child*** of n'. Hence, the first two conditions may be rephrased by saying that the root is the only node that does not have a parent, and every other node has a unique parent.
We shall call $n \in N$ a ***leaf*** node (or simply a leaf) iff it has no children.

Definition 2.72. A ***branch*** [8] $\mathscr{B}$ of $\mathscr{T}$ is a path starting at the root that either ends at a leaf, or is infinite.

Remark 2.73. It follows from condition (2) that the path in condition (3) is unique. So, any finite branch is uniquely determined by its leaf node.

We will primarily be interested in trees in which nodes are labelled with prefixed signed bounding implications.

Definition 2.74. A ***labeling*** of a tree $\mathscr{T} = (N, E)$ is any function $U : N \to pSBI$. A ***labeled tree*** is a pair $(\mathscr{T}, U)$ consisting of a tree and a labeling of that tree.
For a path $\mathscr{P}$ in $\mathscr{T}$, we let

$$U(\mathscr{P}) \coloneqq \bigcup_n \{U(n)\},$$

where n runs over the set of nodes in $\mathscr{P}$. We say that $U(\mathscr{P})$ is the set of labels occurring in $\mathscr{P}$.

It will be helpful to represent trees pictorially. We draw a circle with a name inside for each node, such that the circles representing the children of each node n are drawn below the circle representing n. We draw a line segment from the circle representing n down to the circle representing n' iff n' is a child of n. Further, for a labeled tree with the labeling U, for each node n we write $U(n)$ (the label of the node) next to the circle representing n. In cases where a name for a node is not important, we simply write the label without a named circle.

[8] Note that we also use the word 'branch' to refer to the elements of a tableau. The justification for this term overloading will be made explicit in Definition 4.4

Definition 2.75. Let $\mathscr{T} = (N, E)$ be a tree. $\mathscr{T}$ is an ***infinite tree*** iff N is infinite. $\mathscr{T}$ is ***finitely generated*** iff every node has a finite number of children

A fundamental result in graph theory, which is often used in proof theory, is König's Lemma [38] and its special case applied to infinite trees.

Proposition 2.76 (König's Lemma). *Every infinite, finitely generated tree must contain at least one infinite branch.*

Proof. See [50, p. 32]. □

3 Unprefixed Tableaux

In [24], Fitting studies the unprefixed tableau system

$$\mathcal{C}\mathbf{K}^{\mathcal{H}} = \{\bot_1, \bot_2, \bot_3, \bot_4, \bot_5, F{\geq}, T{\geq}, F{\leq}, T{\leq}, T\wedge, F\wedge, T\vee, F\vee, T{\supset}, F{\supset}, \mathbf{K}F\Box, \mathbf{K}F\Diamond\},$$

where these rules are:

$$(\bot_1)\ \frac{X; T(\underline{a} \supset \underline{b})}{\bot} \qquad \text{Where } a \not\leq b$$

$$(\bot_2)\ \frac{X; F(\underline{a} \supset \underline{b})}{\bot} \qquad \text{Where } a \leq b$$

$$(\bot_3)\ \frac{X; F(\underline{0} \supset \varphi)}{\bot}$$

$$(\bot_4)\ \frac{X; F(\varphi \supset \underline{1})}{\bot}$$

$$(\bot_5)\ \frac{X; T(\underline{b} \supset \varphi); F(\underline{a} \supset \varphi)}{\bot} \qquad \text{Where } a \leq b$$

$$(F{\geq})\ \frac{X; F(\underline{a} \supset \varphi)}{X; T(\varphi \supset \underline{t_1}) \ \big|\ \dots \ \big|\ X; T(\varphi \supset \underline{t_n})}$$

Where $t_1, \dots, t_n$ are all the maximal $\mathcal{H}$-truth values not above a, and $a \neq 0$.

$$(T{\geq})\ \frac{X; T(\underline{a} \supset \varphi)}{X; F(\varphi \supset \underline{t_i})}$$

Where t_i is any maximal $\mathcal{H}$-truth value not above a, and $a \neq 0$.

$$(F{\leq})\ \frac{X; F(\varphi \supset \underline{a})}{X; T(\underline{u_1} \supset \varphi) \ \big|\ \dots \ \big|\ X; T(\underline{u_k} \supset \varphi)}$$

Where $u_1, \dots, u_k$ are all the minimal $\mathcal{H}$-truth values not below a, and $a \neq 1$.

$$(T{\leq})\ \frac{X; T(\varphi \supset \underline{a})}{X; F(\underline{u_i} \supset \varphi)}$$

Where u_i is any minimal $\mathcal{H}$-truth value not below a, and $a \neq 1$.

$$(T\wedge)\ \frac{X; T(\underline{a} \supset (\varphi \wedge \psi))}{X; T(\underline{a} \supset \varphi); T(\underline{a} \supset \psi)}$$

Where $a \neq 0$.

$$(F\wedge)\ \frac{X; F(\underline{a} \supset (\varphi \wedge \psi))}{X; F(\underline{a} \supset \varphi) \quad \mid \quad X; F(\underline{a} \supset \psi)}$$

Where $a \neq 0$.

$$(T\vee)\ \frac{X; T((\varphi \vee \psi) \supset \underline{a})}{X; T(\varphi \supset \underline{a}); T(\psi \supset \underline{a})}$$

Where $a \neq 1$.

$$(F\vee)\ \frac{X; F((\varphi \vee \psi) \supset \underline{a})}{X; F(\varphi \supset \underline{a}) \quad \mid \quad X; F(\psi \supset \underline{a})}$$

Where $a \neq 1$.

$$(F{\supset})\ \frac{X; F(\underline{a} \supset (\varphi \supset \psi))}{X; T(\underline{t_1} \supset \varphi); F(\underline{t_1} \supset \psi) \quad \mid \quad \ldots \quad \mid \quad X; T(\underline{t_n} \supset \varphi); F(\underline{t_n} \supset \psi)}$$

Where $a \neq 0$ and $t_1, \ldots, t_n$ are all the $\mathcal{H}$-truth values below a except 0.

$$(T{\supset})\ \frac{X; T(\underline{a} \supset (\varphi \supset \psi))}{X; F(\underline{t_i} \supset \varphi) \quad \mid \quad X; T(\underline{t_i} \supset \psi)}$$

Where $a \neq 0$ and t_i is any $\mathcal{H}$-truth value below a except 0.

$$(\mathbf{K}F\Box)\ \frac{X; F(\underline{a} \supset \Box\varphi)}{X^{\#}(t_1); F(\underline{a \wedge t_1} \supset \varphi) \quad \mid \quad \ldots \quad \mid \quad X^{\#}(t_n); F(\underline{a \wedge t_n} \supset \varphi)}$$

Where $t_1, \ldots, t_n$ are all the $\mathcal{H}$-truth values s.t. $a \wedge t_i \neq 0$.

$$(\mathbf{K}F\Diamond)\ \frac{X; F(\Diamond\varphi \supset a)}{X^{\#}(t_1); F(\varphi \supset \underline{t_1 \Rightarrow a}) \quad \mid \quad \ldots \quad \mid \quad X^{\#}(t_n); F(\varphi \supset \underline{t_n \Rightarrow a})}$$

Where $t_1, \ldots t_n$ are all the $\mathcal{H}$-truth values s.t. $t_i \Rightarrow a \neq 1$.

We call $\bot_1, \bot_2, \bot_3, \bot_4, \bot_5$ the ***closure rules***; $F{\geq}, T{\geq}, F{\leq}, T{\leq}$ the ***reversal rules***; and $\mathbf{K}F\Box, \mathbf{K}F\Diamond$ the ***modal rules***.

Remark 3.1. The side conditions of the rules play varying roles. Most importantly, they may be essential for proving soundness and completeness. For instance, note how, in the following proof of Lemma 3.2, the side conditions of the reversal rules bring Lemmas 2.9 and 2.11 into play. Or consider how the side conditions of the closure rules play a role in proving Lemma 3.4 (1). On the other hand, a side condition may not be essential for soundness and completeness, but rather provide convenience by preventing the introduction of superfluous branches. For example, the side condition of $F\vee$ prevents the introduction of branches that could be immediately closed via an application of $\bot_4$. Similarly, part of the side condition of $\mathbf{K}F\Box$ prevents the introduction of branches that could be immediately closed via an application of $\bot_3$.

In the modal rules, for any set of signed bounding implications X we define

$$X^{\#}(c) := \{T(\underline{a \wedge c} \supset \theta) \mid T(\underline{a} \supset \Box\theta) \in X \text{ and } a \wedge c \neq 0\} \cup \{T(\theta \supset \underline{c \Rightarrow a}) \mid T(\Diamond\theta \supset \underline{a}) \in X \text{ and } c \Rightarrow a \neq 1\}.$$

This is the many-valued analog of the '# operation' present in the modal rules of Fitting's tableau system for standard modal logic (see Chapter 2 of [20]). Its definition aims to capture the following idea: If the formulas in X are true at some world $\mathfrak{s}$ in an $\mathcal{H}$-model, and we move from that world to a generic alternative world $\mathfrak{v}$ that is accessible from $\mathfrak{s}$ to degree c, then $X^{\#}(c)$ contains all the formulas that we can infer must hold at $\mathfrak{v}$. Indeed, this idea is proven for Lemma 3.2.

Fitting proves the soundness and completeness of $\mathcal{C}\mathbf{K}^{\mathcal{H}}$ with respect to the class of all $\mathcal{H}$-frames. Some results proven there will be reused to prove results about the new tableau systems we introduce. So, to make this work self-contained, these results will be restated here.
One such result is relevant to the soundness of all our systems, and the arguments that constitute its proof are scattered throughout [24]. We present it in consolidated form below.

Lemma 3.2. *Let $\mathcal{F}$ be an arbitrary class of $\mathcal{H}$-frames. ρ preserves $\mathcal{F}$-satisfiablility for every $\rho \in \mathcal{C}\boldsymbol{K}^{\mathcal{H}}$.*

Proof. The arguments in [24] are presented under the assumption that $\mathcal{F}$ is the class of all $\mathcal{H}$-frames. The main motivation for presenting this Lemma and its proof is to highlight that, in fact, no assumptions need to be placed on the class $\mathcal{F}$. Hence, we can make free use of this lemma in the context of the new tableau systems to come, which inherit many of their rules from $\mathcal{C}\mathbf{K}^{\mathcal{H}}$.
Let $\rho \in \mathcal{C}\mathbf{K}^{\mathcal{H}}$ and T a $\mathcal{F}$-satisfiable $\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau. Suppose T^* was derived from T via an application of ρ. That is, $T^* = f(T)$ for some f described by ρ. We must show that T^* is $\mathcal{F}$-satisfiable. Since T is $\mathcal{F}$-satisfiable, there exists some branch $S \in T$ that is $\mathcal{F}$-satisfiable. That is to say, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$ and world $\mathfrak{s} \in W$ s.t. $(W, R) \in \mathcal{F}$ and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. If $S \in T^*$, then clearly T^* is $\mathcal{F}$-satisfiable. Indeed, this must be the case if $\rho \in \{\bot_1, \bot_2, \bot_3, \bot_4, \bot_5\}$ (i.e., ρ is a closure rule), for any instantiation of the numerator of these rules is not $\mathcal{F}$-satisfiable, but S is $\mathcal{F}$-satisfiable by assumption. Hence, any application of a closure rule to T cannot be on S and so will not lead to S being replaced in T^*. Now, without loss of generality, assume $S \notin T^*$ (and so also $\rho \notin \{\bot_1, \bot_2, \bot_3, \bot_4, \bot_5\}$). Then we must have that S is an instantiation of the numerator of ρ and the corresponding instantiation(s) of the denominator(s) is (are) in T^*. We proceed to show that the corresponding instantiations of at least one of the denominators is $\mathcal{F}$-satisfiable. There are several cases depending on which rule ρ is.

- $\rho = F\geq$
  So S is an instantiation of $X; F(\underline{a} \supset \varphi)$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset \varphi) \neq 1$. Hence, $a \nleq V(\mathfrak{s}, \varphi)$. Let $G := \{c \in H \mid c \nleq a\} \subseteq H$. G is finite since H is finite, and $V(\mathfrak{s}, \varphi) \in G$. So, by Lemma 2.9 (1), we have $V(\mathfrak{s}, \varphi) \leq t_i$ for some $t_i \in max(G)$. Or equivalently, $V(\mathfrak{s}, \varphi \supset \underline{t_i}) = 1$. Thus, each member of $X; T(\varphi \supset \underline{t_i})$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. I.e., $X; T(\varphi \supset \underline{t_i})$ is $\mathcal{F}$-satisfiable for some $t_i \in max(\{c \in H \mid c \nleq a\})$.

- $\rho = T\geq$
  So S is an instantiation of $X; T(\underline{a} \supset \varphi)$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset \varphi) = 1$. Hence, $a \leq V(\mathfrak{s}, \varphi)$. Let $t_i \in max(\{c \in H \mid c \nleq a\})$. By Lemma 2.11 (1), we have

$V(\mathfrak{s}, \varphi) \not\leq t_i$. Or equivalently, $V(\mathfrak{s}, \varphi \supset \underline{t_i}) \neq 1$. Thus, each member of $X; T(\varphi \supset \underline{t_i})$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. I.e., $X; T(\varphi \supset \underline{t_i})$ is $\mathcal{F}$-satisfiable for each $t_i \in max(\{c \in H \mid c \not\leq a\})$.

- $\rho \in \{F\leq, T\leq\}$
  The arguments are similar to those for the previous two cases, except we use the dual facts concerning maximal and minimal elements in a partially ordered set.

- $\rho = T\wedge$
  So S is an instantiation of $X; T(\underline{a} \supset (\varphi \wedge \psi))$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset (\varphi \wedge \psi)) = 1$. Hence,

$$\begin{aligned} a &\leq V(\mathfrak{s}, \varphi \wedge \psi) \\ &= V(\mathfrak{s}, \varphi) \wedge V(\mathfrak{s}, \psi). \end{aligned}$$

  So, $a \leq V(\mathfrak{s}, \varphi)$ and $a \leq V(\mathfrak{s}, \psi)$. Or equivalently, $V(\mathfrak{s}, \underline{a} \supset \varphi) = 1$ and $V(\mathfrak{s}, \underline{a} \supset \psi) = 1$. Thus, each member of $X; T(\underline{a} \supset \varphi); T(\underline{a} \supset \psi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. I.e., $X; T(\underline{a} \supset \varphi); T(\underline{a} \supset \psi)$ is $\mathcal{F}$-satisfiable.

- $\rho = F\wedge$
  So S is an instantiation of $X; F(\underline{a} \supset (\varphi \wedge \psi))$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset (\varphi \wedge \psi)) \neq 1$. Hence,

$$\begin{aligned} a &\not\leq V(\mathfrak{s}, \varphi \wedge \psi) \\ &= V(\mathfrak{s}, \varphi) \wedge V(\mathfrak{s}, \psi). \end{aligned}$$

  So, $a \not\leq V(\mathfrak{s}, \varphi)$ or $a \not\leq V(\mathfrak{s}, \psi)$. Equivalently, $V(\mathfrak{s}, \underline{a} \supset \varphi) \neq 1$ or $V(\mathfrak{s}, \underline{a} \supset \psi) \neq 1$. Thus, each member of $X; F(\underline{a} \supset \varphi)$ or each member of $X; F(\underline{a} \supset \psi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$.

- $\rho \in \{T\vee, F\vee\}$
  The arguments are similar to those for the previous two cases, except we use the dual facts concerning joins in a lattice.

- $\rho = F\supset$
  So S is an instantiation of $X; F(\underline{a} \supset (\varphi \supset \psi))$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset (\varphi \supset \psi)) \neq 1$. Hence,

$$\begin{aligned} a &\not\leq V(\mathfrak{s}, \varphi \supset \psi) \\ &= V(\mathfrak{s}, \varphi) \Rightarrow V(\mathfrak{s}, \psi). \end{aligned}$$

  Equivalently, $(a \wedge V(\mathfrak{s}, \varphi)) \not\leq V(\mathfrak{s}, \psi)$ (Recall Proposition). Consider $t_i = a \wedge V(\mathfrak{s}, \varphi)$. Clearly we have $t_i \leq a$, $t_i \neq 0$, $t_i \leq V(\mathfrak{s}, \varphi)$ (i.e., $V(\mathfrak{s}, \underline{a} \supset \varphi) = 1$) and $t_i \not\leq V(\mathfrak{s}, \psi)$ (i.e., $V(\mathfrak{s}, \underline{a} \supset \psi) \neq 1$). Thus, each member of $X; T(\underline{a} \supset \varphi); F(\underline{a} \supset \psi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$.

- $\rho = T\supset$
  So S is an instantiation of $X; T(\underline{a} \supset (\varphi \supset \psi))$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset (\varphi \supset \psi)) = 1$. Hence,

$$\begin{aligned} a &\leq V(\mathfrak{s}, \varphi \supset \psi) \\ &= V(\mathfrak{s}, \varphi) \Rightarrow V(\mathfrak{s}, \psi), \end{aligned}$$

and so $(a \wedge V(\mathfrak{s}, \varphi)) \leq V(\mathfrak{s}, \psi)$. Let t_i be any $\mathcal{H}$-truth value below a except 0. Assuming $t_i \leq V(\mathfrak{s}, \varphi)$, then $t_i \leq a \wedge V(\mathfrak{s}, \varphi) \leq V(\mathfrak{s}, \psi)$.
Thus, we have either $t_i \not\leq V(\mathfrak{s}, \varphi)$ (so $F(\underline{t_i} \supset \varphi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$)
or $t_i \leq V(\mathfrak{s}, \varphi)$, in which case $t_i \leq V(\mathfrak{s}, \psi)$ (so $T(\underline{t_i} \supset \psi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$).

- $\rho = \mathbf{K}F\Box$
So S is an instantiation of $X; F(\underline{a} \supset \Box\varphi)$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset \Box\varphi) \neq 1$. Hence,

$$\begin{aligned} a &\not\leq V(\mathfrak{s}, \Box\varphi) \\ &= \bigwedge\{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow V(\mathfrak{v}, \varphi) \mid \mathfrak{v} \in W\}. \end{aligned}$$

So, there must exist some $\mathfrak{v}_0 \in W$ s.t. $a \not\leq R(\mathfrak{s}, \mathfrak{v}_0) \Rightarrow V(\mathfrak{v}_0, \varphi)$. Consider $t_i = R(\mathfrak{s}, \mathfrak{v}_0)$; then we have $a \wedge t_i \not\leq V(\mathfrak{v}_0, \varphi)$, and hence $F(\underline{a \wedge t_i} \supset \varphi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{v}_0$.

We proceed to show that $X^{\#}(t_i)$ is satisfied at $\mathfrak{v}_0$. Let $x \in X^{\#}(t_i)$. First, assume x is of the form $T(\underline{a \wedge t_i} \supset \theta)$ where $T(\underline{a} \supset \Box\theta) \in X$. Since X is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$, we have that $V(\mathfrak{s}, \underline{a} \supset \Box\theta) = 1$. Hence,

$$\begin{aligned} a &\leq V(\mathfrak{s}, \Box\theta) \\ &= \bigwedge\{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow V(\mathfrak{v}, \theta) \mid \mathfrak{v} \in W\} \\ &\leq R(\mathfrak{s}, \mathfrak{v}_0) \Rightarrow V(\mathfrak{v}_0, \theta) && (\mathfrak{v}_0 \in W) \\ &= t_i \Rightarrow V(\mathfrak{v}_0, \theta) \end{aligned}$$

Therefore, $a \wedge t_i \leq V(\mathfrak{v}_0, \theta)$ and so $T(\underline{a \wedge t_i \wedge d} \supset \theta)$ is satisfied at $\mathfrak{v}_0$.

Now, assume x is of the form $T(\theta \supset \underline{t_i \Rightarrow a})$, where $T(\Diamond\theta \supset \underline{a}) \in X$. Since X is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$, we have that $V(\mathfrak{s}, \Diamond\theta \supset \underline{a}) = 1$. Hence,

$$\begin{aligned} a &\geq V(\mathfrak{s}, \Diamond\theta) \\ &= \bigvee\{R(\mathfrak{s}, \mathfrak{v}) \wedge V(\mathfrak{v}, \theta) \mid \mathfrak{v} \in W\} \\ &\geq R(\mathfrak{s}, \mathfrak{v}_0) \wedge V(\mathfrak{v}_0, \theta) && (\mathfrak{v}_0 \in W) \\ &= t_i \wedge V(\mathfrak{v}_0, \theta). \end{aligned}$$

Therefore, $V(\mathfrak{v}_0, \theta) \leq t_i \Rightarrow a$ and so $T(\theta \supset \underline{(t_i \wedge d) \Rightarrow a})$ is satisfied at $\mathfrak{v}_0$.
And so, the entire instantiation of the denominator $X^{\#}(t_i); F(\underline{a \wedge t_i} \supset \varphi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{v}_0$

- $\rho = \mathbf{K}F\Diamond$
So S is an instantiation of $X; F(\Diamond\varphi \supset \underline{a})$, and each member of S is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \Diamond\varphi \supset \underline{a}) \neq 1$. Hence,

$$\begin{aligned} a &\not\geq V(\mathfrak{s}, \Diamond\varphi) \\ &= \bigvee\{R(\mathfrak{s}, \mathfrak{v}) \wedge V(\mathfrak{v}, \varphi) \mid \mathfrak{v} \in W\}. \end{aligned}$$

So, there must exist some $\mathfrak{v}_0 \in W$ s.t. $R(\mathfrak{s}, \mathfrak{v}_0) \wedge V(\mathfrak{v}_0, \varphi) \not\leq a$. Let $R(\mathfrak{s}, \mathfrak{v}_0) = t_i$; then we have $V(\mathfrak{v}_0, \varphi) \not\leq t_i \Rightarrow a$, and hence $F(\varphi \supset \underline{t_i \Rightarrow a})$ is satisfied by $\mathfrak{M}$ at $\mathfrak{v}_0$.
To show that $X^{\#}(t_i)$ is satisfied at $\mathfrak{v}_0$, we employ the same argument as in the preceding case.

□

Remark 3.3. Note that in the above proof, we explicitly distinguish between numerators/denominators of a rule and the instantiations that they describe. When arguing in our metasystem about general instantiations, this becomes cumbersome and quite awkward. As such, in the rest of this work we will usually not make such distinctions, and conflate the numerator/denominator with its instantiation.

This Lemma indicates an important intuition about all the unprefixed tableaux we will deal with: a particular branch of an unprefixed tableau gives us information about a single world in a hypothetical model that could possibly satisfy the branch at that world. Then conceptually, an application of a modal rule to a branch involves moving to another world. And in doing so, we take all that information which we are certain will hold at the new world, while discarding all that information which we are not certain will hold at the new world. This discarding, as performed by the # operation, leads us to say that such rules are *forgetful/destructive*. The use of destructive modal rules limits the applicability of the unprefixed systems to specific logics.

In proving completeness, Fitting makes use of the following constructions, where X is any set of signed bounding implications and φ an arbitrary formula.

$$bound_X(\varphi) \coloneqq \bigvee\{a \mid T(\underline{a} \supset \varphi) \in X\}$$

$$bound^X(\varphi) \coloneqq \bigwedge\{a \mid T(\varphi \supset \underline{a}) \in X\}$$

The following properties of this construction are vital for completeness.

Lemma 3.4. *Let $\mathcal{C}$ be a tableau system which contains the closure rules as well as the reversal rules. For any maximally $\mathcal{C}$-consistent set S, formula φ, and $\mathcal{H}$-truth value a, we have:*

1. $bound_S(\varphi) \leq bound^S(\varphi)$
2. *If* $T(\underline{a} \supset \varphi) \in S$, *then* $a \leq bound_S(\varphi)$
3. *If* $T(\varphi \supset \underline{a}) \in S$, *then* $bound^S(\varphi) \leq a$
4. *If* $F(\underline{a} \supset \varphi) \in S$, *then* $a \not\leq bound^S(\varphi)$
5. *If* $F(\varphi \supset \underline{a}) \in S$, *then* $bound_S(\varphi) \not\leq a$

Proof. This result appears as Lemma 6.4 and Proposition 6.5 in [24]. There, it is assumed that $\mathcal{C} = \mathcal{C}\mathbf{K}^{\mathcal{H}}$. However, the proofs only use the assumption that S is maximally consistent relative to a system containing the closure and reversal rules, and hence the results hold for any system containing these rules.
For instance, suppose $F(\varphi \supset \underline{a}) \in S$ but $bound_S(\varphi) \leq a$. Since S is $\mathcal{C}$-consistent and $\mathcal{C}$ contains the reversal rule $F\leq$, we must have that $S \cup \{T(\underline{u_i} \supset \varphi)\}$ is $\mathcal{C}$-consistent for some $u_i \not\leq a$. Thus, since S is *maximally* $\mathcal{C}$-consistent, we must have $T(\underline{u_i} \supset \varphi) \in S$, and hence $u_i \leq bound_S(\varphi)$ by (2). But this contradicts $bound_S(\varphi) \leq a$ and $u_i \not\leq a$. □

We now go on to introduce new unprefixed tableau systems by expanding and modifying $\mathcal{C}\mathbf{K}^{\mathcal{H}}$. All our unprefixed tableau systems use some version of destructive # operations in their modal rules. Essentially, this means these systems can only capture logics where the frames do not require "complex" interactions between successors of a world or between a world and its predecessors. Logics with (many-valued versions) of the tree-model property like **K**, **K4** and **KT** are quintessential examples. Logics that require frames with many-valued versions of properties like symmetry or confluence are not easily amenable to treatment with unprefixed tableaux. Indeed, when constructing satisfying models based on these frames, extra "bookkeeping" is needed to relate newly added worlds to previously added worlds. The prefixed tableau systems we introduce in Section 4 lend themselves well to this task. We leave as future work the task of formally characterize the limitations of the unprefixed tableau systems; possibly in terms of a suitable generalization of the tree-model property.

3.1 Tableau System for $\mathbf{T}_d^{\mathcal{H}}$

Fix some arbitrary $d \in H$. We now proceed to give a tableau system which is sound and complete with respect to the class of all d-reflexive $\mathcal{H}$-frames.
Define the following tableau rules.

$$(\mathbf{T}T\Box_d)\ \frac{X; T(\underline{a} \supset \Box\varphi)}{X; T(\underline{a \wedge d} \supset \varphi)}$$

$$(\mathbf{T}T\Diamond_d)\ \frac{X; T(\Diamond\varphi \supset \underline{a})}{X; T(\varphi \supset \underline{d \Rightarrow a})}$$

Consider the unprefixed tableau system

$$\mathcal{C}\mathbf{T}_d^{\mathcal{H}} := \mathcal{C}\mathbf{K}^{\mathcal{H}} \cup \{\mathbf{T}T\Box_d, \mathbf{T}T\Diamond_d\}.$$

Example 3.5. For this example, let us take $\mathcal{H} = \mathcal{H}^3$ and $d = 1$. We proceed to construct a $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-tableau for $\{F(\underline{1} \supset \Diamond\underline{1})\}$. Start with the base $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-tableau:

$$T_0 = \{ \{F(\underline{1} \supset \Diamond\underline{1})\} \}.$$

Applying $F{\geq}$ to $\{F(\underline{1} \supset \Diamond\underline{1})\} \in T_0$, we derive

$$T_1 = \{ \{T(\Diamond\underline{1} \supset \underline{h})\} \}.$$

Applying $\mathbf{T}T\Diamond_d$ to $\{T(\Diamond\underline{1} \supset \underline{h})\} \in T_1$, we derive

$$T_2 = \{ \{T(\underline{1} \supset \underline{d \Rightarrow h})\} \},$$

where $\underline{d \Rightarrow h} \equiv \underline{h}$. So, applying $\perp_1$ to $\{T(\underline{1} \supset \underline{d \Rightarrow h})\} \in T_2$, we derive

$$T_3 = \{ \{\perp\} \},$$

which is a closed $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-tableau for $\{F(\underline{1} \supset \Diamond\underline{1})\}$. I.e. T_3 is a $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-proof of $\Diamond\underline{1}$, and hence $\Diamond\underline{1}$ is a theorem of $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$. Further, it can be easily argued semantically that $\Diamond\underline{1}$ is $\mathsf{Refl}_d^{\mathcal{H}}$-valid. This alignment of $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-theoremhood and $\mathsf{Refl}_d^{\mathcal{H}}$-validity is no coincidence. Rather, it is a direct consequence of the soundness and completeness of $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ wrt $\mathsf{Refl}_d^{\mathcal{H}}$, which we proceed to prove now.

Lemma 3.6. *ρ preserves $\mathsf{Refl}_d^{\mathcal{H}}$-satisfiablility for every $\rho \in \mathcal{C}\mathbf{T}_d^{\mathcal{H}}$.*

Proof. Let $\rho \in \mathcal{C}\mathbf{T}_d^{\mathcal{H}}$. If $\rho \in \mathcal{C}\mathbf{K}^{\mathcal{H}}$, then the result follows from Lemma 3.2 with $\mathcal{F} = \mathsf{Refl}_d^{\mathcal{H}}$. We consider the newly introduced rules now, which do rely on the d-reflexivity of the satisfying models. Let $\rho \in \{\mathbf{T}T\Box_d, \mathbf{T}T\Diamond_d\}$, and suppose that the numerator $\mathcal{N}$ of ρ is $\mathsf{Refl}_d^{\mathcal{H}}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W,R),V)$ and world $\mathfrak{s} \in W$ s.t. $(W,R) \in \mathsf{Refl}_d^{\mathcal{H}}$ and each member of $\mathcal{N}$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. We wish to show that the denominator $\mathcal{D}$ is $\mathsf{Refl}_d^{\mathcal{H}}$-satisfiable.

- $\rho = \mathbf{T}T\Box_d$

$$\mathcal{N} = X; T(\underline{a} \supset \Box\varphi).$$

So $T(\underline{a} \supset \Box\varphi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. That is, $V(\mathfrak{s}, \underline{a} \supset \Box\varphi) = 1$. Hence,

$$\begin{aligned} a &\leq V(\mathfrak{s}, \Box\varphi) \\ &= \bigwedge\{R(\mathfrak{s},\mathfrak{v}) \Rightarrow V(\mathfrak{v},\varphi) \mid \mathfrak{v} \in W\} \\ &\leq R(\mathfrak{s},\mathfrak{s}) \Rightarrow V(\mathfrak{s},\varphi) \\ &\leq d \Rightarrow V(\mathfrak{s},\varphi). && (R(\mathfrak{s},\mathfrak{s}) \geq d \text{ since } (W,R) \in \mathsf{Refl}_d^{\mathcal{H}}) \end{aligned}$$

But $a \leq d \Rightarrow V(\mathfrak{s},\varphi)$ iff $(a \wedge d) \leq V(\mathfrak{s},\varphi)$. So $(a \wedge d) \Rightarrow V(\mathfrak{s},\varphi) = V(\mathfrak{s}, \underline{a \wedge d} \supset \varphi) = 1$. Thus, $\mathcal{D} = X; T(\underline{a \wedge d} \supset \varphi)$ is $\mathsf{Refl}_d^{\mathcal{H}}$-satisfiable.

- $\rho = \mathbf{T}T\Diamond_d$

$$\mathcal{N} = X; T(\Diamond\varphi \supset \underline{a}).$$

So $V(\mathfrak{s}, \Diamond\varphi \supset \underline{a}) = 1$. Hence,

$$\begin{aligned} a &\geq V(\mathfrak{s}, \Diamond\varphi \supset \underline{a}) \\ &= \bigvee\{R(\mathfrak{s},\mathfrak{v}) \wedge V(\mathfrak{v},\varphi) \mid \mathfrak{v} \in W\} \\ &\geq R(\mathfrak{s},\mathfrak{s}) \wedge V(\mathfrak{s},\varphi) \\ &\geq d \wedge V(\mathfrak{s},\varphi). && (R(\mathfrak{s},\mathfrak{s}) \geq d \text{ since } (W,R) \in \mathsf{Refl}_d^{\mathcal{H}}) \end{aligned}$$

But $d \wedge V(\mathfrak{s},\varphi) \leq a$ iff $V(\mathfrak{s},\varphi) \leq (d \Rightarrow a)$. So $V(\mathfrak{s},\varphi) \Rightarrow (d \Rightarrow a) = V(\mathfrak{s}, \varphi \supset \underline{d \Rightarrow a}) = 1$. Thus, $\mathcal{D} = X; T(\varphi \supset \underline{d \Rightarrow a})$ is $\mathsf{Refl}_d^{\mathcal{H}}$-satisfiable.

□

Proposition 3.7. *$\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ is sound with respect to $\mathsf{Refl}_d^{\mathcal{H}}$.*

Proof. By Proposition 2.64, the soundness of $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ follows from the previous Lemma. □

Proposition 3.8. *$\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ is strongly complete with respect to $\mathsf{Refl}_d^{\mathcal{H}}$.*

Proof. Define an $\mathcal{H}$-frame (W, R) where W is the set of all maximally $\mathcal{C}\mathbf{K}^{\mathcal{H}}$-consistent sets of signed bounding implications and R is given by

$$R(X,Y) = \bigwedge\{bound_X(\Box\varphi) \Rightarrow bound_Y(\varphi) \mid \varphi \text{ a formula}\}$$
$$\wedge \bigwedge\{bound^Y(\varphi) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}.$$

We first argue that (W, R) is d-reflexive. Let $X \in W$. We wish to show that

$$R(X,X) = \bigwedge\{bound_X(\Box\varphi) \Rightarrow bound_X(\varphi) \mid \varphi \text{ a formula}\}$$
$$\wedge \bigwedge\{bound^X(\varphi) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}$$
$$\geq d.$$

By Proposition 2.17.2, this is the case iff for every formula φ, $d \leq bound_X(\Box\varphi) \Rightarrow bound_X(\varphi)$ and $d \leq bound^X(\varphi) \Rightarrow bound^X(\Diamond\varphi)$. We proceed to show that both conditions hold.

- $d \leq bound_X(\Box\varphi) \Rightarrow bound_X(\varphi)$ iff $d \wedge bound_X(\Box\varphi) \leq bound_X(\varphi)$ iff $\bigvee\{(d \wedge a) \mid T(\underline{a} \supset \Box\varphi) \in X\} \leq bound_X(\varphi)$ iff for all a s.t. $T(\underline{a} \supset \Box\varphi) \in X$, $(a \wedge d) \leq bound_X(\varphi)$.
  Consider an arbitrary a s.t. $T(\underline{a} \supset \Box\varphi) \in X$. Since X is $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistent, $X \cup \{T(\underline{a \wedge d} \supset \varphi)\}$ must be $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistent. For assume otherwise. Then there exists some finite subset X' of X s.t. a $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-tableau for $X' \cup \{T(\underline{a \wedge d} \supset \varphi)\}$ is closed. But then, by using the $\mathbf{T}T\Box_d$ rule, we can construct a $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-tableau for $X' \cup \{T(\underline{a} \supset \Box\varphi)\}$ which is closed, contradicting the $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistency of X. So $X \cup \{T(\underline{a \wedge d} \supset \varphi)\}$ is $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistent, and since X is maximal we can conclude that $T(\underline{a \wedge d} \supset \varphi) \in X$. Thus, by the definition of $bound_X$, $(a \wedge d) \leq bound_X(\varphi)$.

- $d \leq bound^X(\varphi) \Rightarrow bound^X(\Diamond\varphi)$ iff $d \wedge bound^X(\varphi) \leq bound^X(\Diamond\varphi)$ iff $d \wedge bound^X(\varphi) \leq \bigwedge\{a \mid T(\Diamond\varphi \supset \underline{a}) \in X\}$ iff for all a s.t. $T(\Diamond\varphi \supset \underline{a}) \in X$, $d \wedge bound^X(\varphi) \leq a$.
  Consider an arbitrary a s.t. $T(\Diamond\varphi \supset \underline{a}) \in X$. By the maximal $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistency of X and rule $\mathbf{T}T\Diamond_d$, we must have that $T(\varphi \supset \underline{d \Rightarrow a}) \in X$. Thus, by the definition of $bound^X$, $bound^X(\varphi) \leq d \Rightarrow a$. So, $d \wedge bound^X(\varphi) \leq a$.

Now consider a model $((W, R), V)$ where V is any valuation s.t. for every $X \in W$ and propositional variable p,

$$bound_X(p) \leq V(X,p) \leq bound^X(p).$$

Such a valuation exists by 3.4 (1).
Supposing the following holds for every formula φ and $X \in W$,

$$bound_X(\varphi) \leq V(X,\varphi) \leq bound^X(\varphi), \tag{3.8.1}$$

then the contrapositive of strong completeness follows easily. For consider an arbitrary set of formulas $\Gamma \cup \{\varphi\}$ and suppose $\Gamma \nvdash_{\mathcal{C}\mathbf{T}^{\mathcal{H}}_d} \varphi$. I.e., there is no finite subset $\Gamma_0 \subseteq \Gamma$ for which there exists a closed $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-tableau for $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma_0\} \cup \{F(\underline{1} \supset \varphi)\}$. So $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma\} \cup \{F(\underline{1} \supset \varphi)\}$ is $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistent, and by Lemma 2.68, we can extend it to a maximal $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-consistent set of signed bounding implications X. By (3.8.1), $bound_X(\varphi) \leq V(X,\varphi) \leq bound^X(\varphi)$. But since $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma\} \subseteq X$, by Lemma 3.4 (2) we have $1 \leq bound_X(\gamma)$ for each $\gamma \in \Gamma$. Hence, $1 \leq V(X,\gamma)$ (i.e., $\mathfrak{M}, X \Vdash \gamma$) for each $\gamma \in \Gamma$. Similarly, since $F(\underline{1} \supset \varphi) \in X$, by Lemma 3.4 (4)we have $1 \not\leq bound^X(\varphi)$. Hence, $1 \not\leq V(X,\varphi)$ (i.e., $\mathfrak{M}, X \nVdash \gamma$). And since $(W, R) \in \mathsf{Refl}^{\mathcal{H}}_d$, we conclude that φ is not $\mathsf{Refl}^{\mathcal{H}}_d$-entailed by Γ.

So, our goal now is to prove that (3.8.1) does in fact hold for every formula φ and $X \in W$. The argument used to do so is exactly the same as that given by Fitting to prove Theorem 9.1 in [24]. The argument only assumes that the system contains all the rules in $\mathcal{C}\mathbf{K}^{\mathcal{H}}$, and this is indeed the case for $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$. To see an outline of the argument, see the proof of Proposition 3.11 in which some steps of the argument are altered from the original proof to accommodate for an alteration of the modal rules.

□

From strong Completeness, we get the following compactness theorem.

Corollary 3.9. *Let $\Gamma \cup \{\varphi\} \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. Γ $\mathsf{Refl}^{\mathcal{H}}_d$-entails φ iff φ is $\mathsf{Refl}^{\mathcal{H}}_d$-entailed by some finite Γ_0 which is a subset of Γ.*

Proof. For the forward implication, suppose Γ $\mathsf{Refl}^{\mathcal{H}}_d$-entails φ. Then, by the strong completeness of $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$ (Proposition 3.8), we have $\Gamma \vdash_{\mathcal{C}\mathbf{T}^{\mathcal{H}}_d} \varphi$. I.e., there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a closed $\mathcal{C}\mathbf{T}^{\mathcal{H}}_d$-tableau for $\{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma_0\} \cup \{F(\underline{1} \supset \varphi)\}$. Hence, by Lemma 2.63 and 3.6, we must have that $\{T(\underline{1} \supset \gamma_0) \mid \gamma \in \Gamma_0\} \cup \{F(\underline{1} \supset \varphi)\}$ is not $\mathsf{Refl}^{\mathcal{H}}_d$-satisfiable. Or equivalently, we have that Γ_0 $\mathsf{Refl}^{\mathcal{H}}_d$-entails φ.
The converse implication follows immediately from the definition of entailment. □

3.2 Tableau System for $\mathbf{K4}^{\mathcal{H}}_d$

Let $d \in H$ and consider the tableau system:

$$\mathcal{C}\mathbf{K4}^{\mathcal{H}}_d = \{\bot_1, \bot_2, \bot_3, \bot_4, \bot_5, F\geq, T\geq, F\leq, T\leq, T\wedge, F\wedge, T\vee, F\vee, T\supset, F\supset, \mathbf{K4}F\Box_d, \mathbf{K4}F\Diamond_d\},$$

where $\mathbf{K4}F\Box_d, \mathbf{K4}F\Diamond_d$ are as follows.

$$(\mathbf{K4}F\Box_d)\quad \frac{X; F(\underline{a} \supset \Box\varphi)}{X^{\#\mathbf{K4}_d}(t_1);\ F(\underline{a \wedge t_1} \supset \varphi) \quad \Big| \quad \cdots \quad \Big| \quad X^{\#\mathbf{K4}_d}(t_n);\ F(\underline{a \wedge t_1} \supset \varphi)}$$

Where $t_1, \ldots, t_n$ are all the $\mathcal{H}$-truth values s.t. $a \wedge t_i \neq 0$.

$$(\mathbf{K4}F\Diamond_d)\quad \frac{X; F(\Diamond\varphi \supset a)}{X^{\#\mathbf{K4}_d}(t_1);\ F(\varphi \supset \underline{t_1 \Rightarrow a}) \quad \Big| \quad \cdots \quad \Big| \quad X^{\#\mathbf{K4}_d}(t_n);\ F(\varphi \supset \underline{t_n \Rightarrow a})}$$

Where $t_1, \ldots t_n$ are all the $\mathcal{H}$-truth values s.t. $t_i \Rightarrow a \neq 1$.

Where for each set X of signed bounding implications and $c \in H$, $X^{\#\mathbf{K4}_d}(c)$ is an extension of $X^{\#}(c)$, namely,

$$\begin{aligned} X^{\#\mathbf{K4}_d}(c) :=& X^{\#}(c) \cup \{T(\underline{a \wedge c \wedge d} \supset \Box\theta) \mid T(\underline{a} \supset \Box\theta) \in X\} \\ &\cup \\ &\{T(\Diamond\theta \supset \underline{(c \wedge d) \Rightarrow a}) \mid T(\Diamond\theta \supset \underline{a}) \in X\}. \end{aligned}$$

This strengthening of the # operation reflects the fact that assuming d-transitivity allows us to infer more when moving to a generic alternative world. This is made explicit in the proof of the following result.

Proposition 3.10. *$\mathcal{C}\mathbf{K4}^{\mathcal{H}}_d$ is sound with respect to $\mathsf{Trans}^{\mathcal{H}}_d$.*

Proof. By Proposition 2.64 and Lemma 3.2, it suffices to show $\mathbf{K4}F\Box_d$ and $\mathbf{K4}F\Diamond_d$ preserve $\mathsf{Trans}_d^{\mathcal{H}}$-satisfiability.

- $\mathbf{K4}F\Box_d$
  Suppose $X; F(\underline{a} \supset \Box\varphi)$ is $\mathsf{Trans}_d^{\mathcal{H}}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W,R),V)$ and world $\mathfrak{s} \in W$ s.t. $(W,R) \in \mathsf{Trans}_d^{\mathcal{H}}$ and each member of $X; F(\underline{a} \supset \Box\varphi)$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \underline{a} \supset \Box\varphi) \neq 1$ and so $a \not\leq V(\mathfrak{s}, \Box\varphi)$. $V(\mathfrak{s}, \Box\varphi) = \bigwedge\{R(\mathfrak{s},\mathfrak{v}) \Rightarrow V(\mathfrak{v},\varphi) \mid \mathfrak{v} \in W\}$, hence there must exist some $\mathfrak{v}_0 \in W$ s.t. $a \not\leq R(\mathfrak{s},\mathfrak{v}_0) \Rightarrow V(\mathfrak{v}_0,\varphi)$. Consider $t_i = R(\mathfrak{s},\mathfrak{v}_0)$; then we have $a \wedge t_i \not\leq V(\mathfrak{v}_0,\varphi)$, and hence $F(\underline{a \wedge t_i} \supset \varphi)$ is satisfied at $\mathfrak{v}_0$.

  We proceed to show that $X^{\#\mathbf{K4}_d}(t_i)$ is satisfied at $\mathfrak{v}_0$. Let $x \in X^{\#\mathbf{K4}_d}(t_i)$. The case in which $x \in X^{\#}(t_i)$ is treated in Lemma 3.2 with $\mathcal{F} = \mathsf{Trans}_d^{\mathcal{H}}$ and $\rho = F\Box$. We consider the two other cases now. First, assume x is of the form $T(\underline{a \wedge t_i \wedge d} \supset \Box\theta)$ where $T(\underline{a} \supset \Box\theta) \in X$. Since X is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$, we have that $V(\mathfrak{s}, \underline{a} \supset \Box\theta) = 1$. Hence,

$$
\begin{aligned}
a &\leq V(\mathfrak{s}, \Box\theta) \\
&= \bigwedge\{R(\mathfrak{s},\mathfrak{r}) \Rightarrow V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} \\
&\leq \bigwedge\{(d \wedge R(\mathfrak{s},\mathfrak{v}_0) \wedge R(\mathfrak{v}_0,\mathfrak{r})) \Rightarrow V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} && \big((W,R) \in \mathsf{Trans}_d^{\mathcal{H}}\big) \\
&= \bigwedge\{(d \wedge R(\mathfrak{s},\mathfrak{v}_0)) \Rightarrow (R(\mathfrak{v}_0,\mathfrak{r}) \Rightarrow V(\mathfrak{r},\theta)) \mid \mathfrak{r} \in W\} \\
&= (d \wedge t_i) \Rightarrow \bigwedge\{R(\mathfrak{v}_0,\mathfrak{r}) \Rightarrow V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} \\
&= (d \wedge t_i) \Rightarrow V(\mathfrak{v}_0, \Box\theta).
\end{aligned}
$$

  Therefore, $a \wedge t_i \wedge d \leq V(\mathfrak{v}_0, \Box\theta)$ and so $T(\underline{a \wedge t_i \wedge d} \supset \Box\theta)$ is satisfied at $\mathfrak{v}_0$.

  Now, assume x is of the form $T(\Diamond\theta \supset \underline{(t_i \wedge d) \Rightarrow a})$ where $T(\Diamond\theta \supset \underline{a}) \in X$. Since X is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$, we have $V(\mathfrak{s}, \Diamond\theta \supset \underline{a}) = 1$. Hence,

$$
\begin{aligned}
a &\geq V(\mathfrak{s}, \Diamond\theta) \\
&= \bigvee\{R(\mathfrak{s},\mathfrak{r}) \wedge V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} \\
&\geq \bigvee\{(d \wedge R(\mathfrak{s},\mathfrak{v}_0) \wedge R(\mathfrak{v}_0,\mathfrak{r})) \wedge V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} && \big((W,R) \in \mathsf{Trans}_d^{\mathcal{H}}\big) \\
&= (d \wedge t_i) \wedge \bigvee\{R(\mathfrak{v}_0,\mathfrak{r}) \wedge V(\mathfrak{r},\theta) \mid \mathfrak{r} \in W\} \\
&= (d \wedge t_i) \wedge V(\mathfrak{v}_0, \Diamond\theta).
\end{aligned}
$$

  Therefore, $V(\mathfrak{v}_0, \Diamond\theta) \leq (t_i \wedge d) \Rightarrow a$ and so $T(\Diamond\theta \supset \underline{(t_i \wedge d) \Rightarrow a})$ is satisfied at $\mathfrak{v}_0$.

- $\mathbf{K4}F\Diamond_d$
  Suppose $X; F(\Diamond\varphi \supset \underline{a})$ is $\mathsf{Trans}_d^{\mathcal{H}}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W,R),V)$ and world $\mathfrak{s} \in W$ s.t. $(W,R) \in \mathsf{Trans}_d^{\mathcal{H}}$ and each member of $X; F(\Diamond\varphi \supset \underline{a})$ is satisfied by $\mathfrak{M}$ at $\mathfrak{s}$. In particular, $V(\mathfrak{s}, \Diamond\varphi \supset \underline{a}) \neq 1$ and so $V(\mathfrak{s}, \Diamond\varphi) \not\leq a$. $V(\mathfrak{s}, \Diamond\varphi) = \bigvee\{R(\mathfrak{s},\mathfrak{v}) \wedge V(\mathfrak{v},\varphi) \mid \mathfrak{v} \in W\}$, hence there must exist some $\mathfrak{v}_0 \in W$ s.t. $R(\mathfrak{s},\mathfrak{v}_0) \wedge V(\mathfrak{v}_0,\varphi) \not\leq a$. Consider $t_i = R(\mathfrak{s},\mathfrak{v}_0)$; then we have that $V(\mathfrak{v}_0,\varphi) \not\leq t_i \Rightarrow a$, and hence $F(\varphi \supset \underline{t_i \Rightarrow a})$ is satisfied at $\mathfrak{v}_0$.
  Showing that $X^{\#\mathbf{K4}_d}(t_i)$ is satisfied at $\mathfrak{v}_0$ is done in the same way as the previous case.

□

Before proving completeness, we note some motivations for the constructions. In the two-valued setting, when proving completeness of the tableau system with respect to transitive Kripke frames via a Henkin-style argument, the accessibility relation of the canonical model is defined differently than for the minimal and reflexive cases. For the standard modal logics, we look to Fitting's proof in [20, p. 56], where he uses the following uniform notation

ν	ν_0
$T\Box\varphi$	$T\varphi$
$F\Diamond\varphi$	$F\varphi$

There, the binary accessibility relation R_b of the canonical model is defined by $(X, Y) \in R_b$ iff $X\# \subseteq Y$, where $X\#$ is defined differently depending on which tableau system we are considering. For **T** and **K**, we have that the definition is the same: for a set of signed formulas X,

$$X\# \coloneqq \{\nu_0 \mid \nu \in X\}.$$

After some consideration, it can be seen that the many valued R as defined in [24] (see proof of Proposition 3.8) is a generalization of $X\# \subseteq Y$ for $X\#$ as defined above. In particular, note that when $\mathcal{H} = \mathbf{2}$, for any maximal $\mathcal{C}\mathbf{K}^{\mathcal{H}}$-consistent (or $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$-consistent) sets X, Y, $R(X, Y) = 1$ iff

$$(\{T\Box\varphi \mid T(\underline{1} \supset \Box\varphi) \in X\} \cup \{F\Diamond\varphi \mid F(\underline{1} \supset \Diamond\varphi) \in X\})\# \subseteq (\{T\varphi \mid T(\underline{1} \supset \varphi) \in Y\} \cup \{F\varphi \mid F(\underline{1} \supset \varphi) \in Y\}).$$

It makes sense that R is defined the same for $\mathcal{C}\mathbf{K}^{\mathcal{H}}$ and $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ given that R_b is the same for **T** and **K**.

However, for **K4**, Fitting defines $X\#$ as follows

$$X\# \coloneqq \{\nu_0 \mid \nu \in X\} \cup \{\nu \mid \nu \in X\}$$

In light of this, it seems reasonable to extend R along the following lines:

$$\begin{aligned} R(X, Y) &= \bigwedge\{\mathit{bound}_X(\Box\varphi) \Rightarrow \mathit{bound}_Y(\varphi) \mid \varphi \text{ a formula}\} \wedge \bigwedge\{\mathit{bound}_X(\Box\varphi) \Rightarrow \mathit{bound}_Y(\Box\varphi) \mid \varphi \text{ a formula}\} \\ &\quad \wedge \bigwedge\{\mathit{bound}^Y(\varphi) \Rightarrow \mathit{bound}^X(\Diamond\varphi) \mid \varphi \text{ a formula}\} \wedge \bigwedge\{\mathit{bound}^Y(\Diamond\varphi) \Rightarrow \mathit{bound}^X(\Diamond\varphi) \mid \varphi \text{ a formula}\} \\ &= \bigwedge\{\mathit{bound}_X(\Box\varphi) \Rightarrow \big(\mathit{bound}_Y(\varphi) \wedge \mathit{bound}_Y(\Box\varphi)\big) \mid \varphi \text{ a formula}\} \\ &\quad \wedge \bigwedge\{\big(\mathit{bound}^Y(\varphi) \vee \mathit{bound}^Y(\Diamond\varphi)\big) \Rightarrow \mathit{bound}^X(\Diamond\varphi) \mid \varphi \text{ a formula}\} \end{aligned}$$

And this is what we do.

Proposition 3.11. *$\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ is strongly complete with respect to $\mathsf{Trans}_d^{\mathcal{H}}$.*

Proof. As in [24] (and Proposition 3.8), we define a canonical $\mathcal{H}$-model $((W, R), V)$ where W is the set of all maximally $\mathcal{C}\mathbf{K4}^{\mathcal{H}}$-consistent sets of signed bounding implications. However, as discussed above, now we define R differently as

$$\begin{aligned} R(X, Y) = &\bigwedge\{\mathit{bound}_X(\Box\varphi) \Rightarrow \big(\mathit{bound}_Y(\varphi) \wedge (d \Rightarrow \mathit{bound}_Y(\Box\varphi))\big) \mid \varphi \text{ a formula}\} \\ &\wedge \bigwedge\{\big(\mathit{bound}^Y(\varphi) \vee (d \wedge \mathit{bound}^Y(\Diamond\varphi))\big) \Rightarrow \mathit{bound}^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}. \end{aligned}$$

Before proceeding, we argue that $(W,R) \in \mathsf{Trans}_d^{\mathcal{H}}$. Let $X,Y,Z \in W$, we want to show that $d \wedge R(X,Y) \wedge R(Y,Z) \leq R(X,Z)$.
To this end, it suffices to show that

$$
\begin{aligned}
&d \wedge \bigwedge\{bound_X(\Box\varphi) \Rightarrow \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big) \mid \varphi \text{ a formula}\}\\
&\wedge \bigwedge\{bound_Y(\Box\varphi) \Rightarrow \big(bound_Z(\varphi) \wedge (d \Rightarrow bound_Z(\Box\varphi))\big) \mid \varphi \text{ a formula}\}\\
&\leq \bigwedge\{bound_X(\Box\varphi) \Rightarrow \big(bound_Z(\varphi) \wedge (d \Rightarrow bound_Z(\Box\varphi))\big) \mid \varphi \text{ a formula}\}
\end{aligned}
$$

and

$$
\begin{aligned}
&d \wedge \bigwedge\{\big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}\\
&\wedge \bigwedge\{\big(bound^Z(\varphi) \vee (d \wedge bound^Z(\Diamond\varphi))\big) \Rightarrow bound^Y(\Diamond\varphi) \mid \varphi \text{ a formula}\}\\
&\leq \bigwedge\{\big(bound^Z(\varphi) \vee (d \wedge bound^Z(\Diamond\varphi))\big) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}.
\end{aligned}
$$

Take the first inequality. It holds iff for every formula ψ,

$$
\begin{aligned}
&d \wedge \bigwedge\{bound_X(\Box\varphi) \Rightarrow \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big) \mid \varphi \text{ a formula}\}\\
&\wedge \bigwedge\{bound_Y(\Box\varphi) \Rightarrow \big(bound_Z(\varphi) \wedge (d \Rightarrow bound_Z(\Box\varphi))\big) \mid \varphi \text{ a formula}\}\\
&\leq bound_X(\Box\psi) \Rightarrow \big(bound_Z(\psi) \wedge (d \Rightarrow bound_Z(\Box\psi))\big).
\end{aligned}
$$

So consider an arbitrary formula ψ. It is sufficient to show that

$$
\begin{aligned}
&d \wedge [bound_X(\Box\psi) \Rightarrow \big(bound_Y(\psi) \wedge (d \Rightarrow bound_Y(\Box\psi))\big)]\\
&\wedge [bound_Y(\Box\psi) \Rightarrow \big(bound_Z(\psi) \wedge (d \Rightarrow bound_Z(\Box\psi))\big)]\\
&\leq [bound_X(\Box\psi) \Rightarrow \big(bound_Z(\psi) \wedge (d \Rightarrow bound_Z(\Box\psi))\big)].
\end{aligned}
$$

Letting $a' = bound_X(\Box\psi), b' = bound_Y(\psi), c' = bound_Y(\Box\psi), d' = bound_Z(\psi), e' = bound_Z(\Box\psi)$, we see that this follows directly from Lemma 2.25 (1).
Similarly, the second inequality can be shown to follow from Lemma 2.25 (2).

As before, we let V be any mapping s.t. for every $X \in W$ and any propositional variable p,

$$bound_X(p) \leq V(X,p) \leq bound^X(p).$$

We now wish to show that for any formula φ and state $X \in W$,

$$bound_X(\varphi) \leq V(X,\varphi) \leq bound^X(\varphi). \tag{3.11.1}$$

As in the case of $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$, the contrapositive of strong completeness follows easily from (3.11.1).
To prove (3.11.1) holds for any formula φ and state $X \in W$, we use the same argument that Fitting uses in [24]. The steps of the argument remain largely the same, but some steps are changed to account for the altered modal rules and definition of R. The argument is by structural induction on formulas.
For the base case and non-modal inductive cases, the argument remains unchanged and is relatively

straight forward, so we omit it here and refer the reader to Proposition 6.6 in [24].
Dealing with the modal cases is more involved and some details are changed. So, for the sake of thoroughness and self-containment, each step of the argument will be reproduced below.
As our induction hypothesis, we let ψ be an arbitrary formula and assume that for any $X \in W$, (3.11.1) holds for it. Suppose φ is constructed from ψ. We wish to show that for any $X \in W$, (3.11.1) holds for φ. We consider the cases in which the principal connective of φ is $\Box$ or $\Diamond$.

- φ is of the form $\Box\psi$.
  Let X_0 be an arbitrary member of W. We first show that $bound_{X_0}(\varphi) \leq V(X_0, \varphi)$.
  Let $Y \in W$. By the definition of R,

$$\begin{aligned} R(X_0, Y) &\leq bound_{X_0}(\Box\psi) \Rightarrow \big(bound_Y(\psi) \wedge (d \Rightarrow bound_Y(\Box\psi))\big) \\ &\leq bound_{X_0}(\Box\psi) \Rightarrow bound_Y(\psi). \end{aligned}$$

  Hence,

$$bound_{X_0}(\Box\psi) \leq R(X_0, Y) \Rightarrow bound_Y(\psi).$$

  By the induction hypothesis, $bound_Y(\psi) \leq V(Y, \psi)$, and so it follows that

$$bound_{X_0}(\Box\psi) \leq R(X_0, Y) \Rightarrow V(Y, \psi).$$

  Since Y is arbitrary, we can conclude that

$$bound_{X_0}(\Box\psi) \leq \bigwedge\{R(X_0, Y) \Rightarrow V(Y, \psi) \mid Y \in W\} = V(X_0, \Box\psi).$$

  We now show that $V(X_0, \varphi) \leq bound^{X_0}(\varphi)$. I.e. we show that $V(X_0, \Box\psi) \leq c$ for every $c \in H$ where $T(\Box\psi \supset \underline{c}) \in X_0$. Suppose, on the contrary, that there exists some $c \in H$ s.t. $T(\Box\psi \supset \underline{c}) \in X_0$ but $V(X_0, \Box\psi) \not\leq c$ – we derive a contradiction.
  Since $V(X_0, \Box\psi) \not\leq c$, by Lemma 2.11 (2), there is some u_i such that $u_i \leq V(X_0, \Box\psi)$, and $u_i \in min(\{u \in H \mid u \not\leq c\})$. Since $T(\Box\psi \supset \underline{c}) \in X_0$, by rule $T\leq\, \in \mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ and the maximal $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistency of X_0, $F(\underline{u_i} \supset \Box\psi) \in X_0$. Hence, by rule $\mathbf{K4}F\Box_d \in \mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$, for some $t_j \in H$ the set $X_0^{\#\mathbf{K4}_d}(t_j) \cup \{F(\underline{u_i \wedge t_j} \supset \psi)\}$ is $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistent. Extend it to a maximal $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistent set Y_0 (which exists by Lemma 2.68). Then $Y_0 \in W$ and by Lemma 3.4 (4), $u_i \wedge t_j \not\leq bound^{Y_0}(\psi)$. Thus, by the induction hypothesis, $u_i \wedge t_j \not\leq V(Y_0, \psi)$.

  Fitting goes on to prove what he calls the 'subordinate result'. The details require some modifications, which we present now. The 'subordinate result' relevant to $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ states that for any maximal $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistent sets X, Y and any $t_j \in H$, if $X^{\#\mathbf{K4}_d}(t_j) \subseteq Y$, then

$$\begin{aligned} t_j \leq{} & R(X, Y) \\ ={} & \bigwedge\{bound_X(\Box\varphi) \Rightarrow \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big) \mid \varphi \text{ a formula}\} \\ & \wedge \bigwedge\{\big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}. \end{aligned}$$

  To prove this, suppose $X^{\#\mathbf{K4}_d}(t_j) \subseteq Y$. It suffices to show that,
  (1) $t_j \leq \bigwedge\{bound_X(\Box\varphi) \Rightarrow \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big) \mid \varphi \text{ a formula}\}$,
  and (2) $t_j \leq \bigwedge\{\big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \Rightarrow bound^X(\Diamond\varphi) \mid \varphi \text{ a formula}\}$.

1. Let φ be an arbitrary formula. We show $t_j \leq bound_X(\Box\varphi) \Rightarrow \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big)$, or equivalently, that $t_j \wedge bound_X(\Box\varphi) \leq \big(bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))\big)$. By Proposition 2.24.15,

$$t_j \wedge bound_X(\Box\varphi) = \bigvee\{a \wedge t_j \mid T(a \supset \Box\varphi) \in X\}.$$

So consider any $a \in H$ s.t. $T(a \supset \Box\varphi) \in X$. It suffices to show that

$$a \wedge t_j \leq bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi)).$$

Without loss of generality, assume $a \wedge t_j \neq 0$. Then, by definition, $T((a \wedge t_j) \supset \varphi), T((a \wedge t_j \wedge d) \supset \Box\varphi) \in X^{\#\mathbf{K4}_d}(t_j)$. But since $X^{\#\mathbf{K4}_d}(t_j) \subseteq Y$, $T((a \wedge t_j) \supset \varphi), T((a \wedge t_j \wedge d) \supset \Box\varphi) \in Y$. So, $a \wedge t_j \leq bound_Y(\varphi)$, and $a \wedge t_j \wedge d \leq bound_Y(\Box\varphi)$, or equivalently, $a \wedge t_j \leq d \Rightarrow bound_Y(\Box\varphi)$. Thus, $a \wedge t_j \leq bound_Y(\varphi) \wedge (d \Rightarrow bound_Y(\Box\varphi))$.

2. Let φ be an arbitrary formula. We show $t_j \leq \big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \Rightarrow bound^X(\Diamond\varphi)$, or equivalently, that

$$t_j \wedge \big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \leq \bigwedge\{a \mid T(\Diamond\varphi \supset a) \in X\}.$$

Consider any $a \in H$ s.t. $T(\Diamond\varphi \supset a) \in X$. Without loss of generality, assume $t_j \Rightarrow a \neq 1$. Then, by definition, $T(\varphi \supset (t_j \Rightarrow a)), T(\Diamond\varphi \supset ((t_j \wedge d) \Rightarrow a)) \in X^{\#\mathbf{K4}_d}(t_j)$. But since $X^{\#\mathbf{K4}_d}(t_j) \subseteq Y$, we must have $T(\varphi \supset (t_j \Rightarrow a)), T(\Diamond\varphi \supset ((t_j \wedge d) \Rightarrow a)) \in Y$. So, $bound^Y(\varphi) \leq t_j \Rightarrow a$, and $bound^Y(\Diamond\varphi) \leq (t_j \wedge d) \Rightarrow a$, or equivalently, $d \wedge bound^Y(\Diamond\varphi) \leq t_j \Rightarrow a$. Thus, $bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi)) \leq t_j \Rightarrow a$. Hence, $t_j \wedge \big(bound^Y(\varphi) \vee (d \wedge bound^Y(\Diamond\varphi))\big) \leq a$.

We use this result to arrive at a contradiction. By definition, we have that

$$\begin{aligned} V(X_0, \Box\psi) &= \bigwedge\{R(X_0, Y) \Rightarrow V(Y, \psi) \mid Y \in W\} \\ &\leq R(X_0, Y_0) \Rightarrow V(Y_0, \psi). \end{aligned}$$

Then,

$$V(X_0, \Box\psi) \wedge R(X_0, Y_0) \leq V(Y_0, \psi).$$

By the 'subordinate result', $t_j \leq R(X_0, Y_0)$, and by our choice of u_i, we have $u_i \leq V(X_0, \Box\psi)$. But then $u_i \wedge t_j \leq V(Y_0, \psi)$, contradicting the fact that $u_i \wedge t_j \not\leq V(Y_0, \psi)$, established above.

- φ is of the form $\Diamond\psi$.
Let X_0 be an arbitrary member of W. We first show that $V(X_0, \varphi) \leq bound^{X_0}(\Diamond\psi)$.
Let $Y \in W$. By the definition of R,

$$\begin{aligned} R(X_0, Y) &\leq \big(bound^Y(\psi) \vee (d \wedge bound^Y(\Diamond\psi))\big) \Rightarrow bound^{X_0}(\Diamond\psi) \\ &\leq bound^Y(\psi) \Rightarrow bound^{X_0}(\Diamond\psi). \end{aligned}$$

Hence,

$$R(X_0, Y) \wedge bound^Y(\psi) \leq bound^{X_0}(\Diamond\psi).$$

By the induction hypothesis, $V(Y, \psi) \leq bound^Y(\psi)$, and so it follows that

$$R(X_0, Y) \wedge V(Y, \psi) \leq bound^{X_0}(\Diamond\psi)$$

Since Y is arbitrary, we can conclude that

$$V(X_0, \Diamond\psi) = \bigvee\{R(X_0, Y) \wedge V(Y, \psi) \mid Y \in W\} \leq bound^{X_0}(\Diamond\psi).$$

We now show that $bound_{X_0}(\varphi) \leq V(X_0, \varphi)$. I.e. we show that $c \leq V(X_0, \Diamond\psi)$ for every $c \in H$ where $T(\underline{c} \supset \Diamond\psi) \in X_0$. Suppose, on the contrary, that there exists some $c \in H$ s.t. $T(\underline{c} \supset \Diamond\psi) \in X_0$ but $c \not\leq V(X_0, \Diamond\psi)$ – we derive a contradiction.
Since $c \not\leq V(X_0, \Diamond\psi)$, by Lemma 2.11 (1) there is some $u_i \in max(\{u \in H \mid c \not\leq u\})$ such that $u_i \geq V(X_0, \Diamond\psi)$. Since $T(\underline{c} \supset \Diamond\psi) \in X_0$, by rule $T\geq \in \mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ and the maximal $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistency of X_0, $F(\Diamond\psi \supset \underline{u_i}) \in X_0$. Hence, by rule $\mathbf{K4}F\Diamond_d \in \mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$, for some $t_j \in H$ the set $X_0^{\#\mathbf{K4}_d}(t_j) \cup \{F(\psi \supset \underline{t_j \Rightarrow u_i})\}$ is $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistent. Extend it to a maximal $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$-consistent set Y_0 (which exists by Lemma 2.68). Then $Y_0 \in W$ and by Lemma 3.4 (5), $bound_{Y_0}(\psi) \not\leq (t_j \Rightarrow u_i)$.
By the subordinate result, $t_j \leq R(X_0, Y_0)$. So,

$$\begin{aligned} t_j \wedge V(Y_0, \psi) &\leq R(X_0, Y_0) \wedge V(Y_0, \psi) \\ &\leq \bigvee\{R(X_0, Y) \wedge V(Y_0, \psi) \mid Y \in W\} \\ &= V(X_0, \Diamond\psi) \\ &\leq u_i. \end{aligned}$$

Thus, $V(Y_0, \psi) \leq (t_j \Rightarrow u_i)$. By the induction hypothesis, $bound_{Y_0}(\psi) \leq V(Y_0, \psi)$, hence

$$bound_{Y_0}(\psi) \leq (t_j \Rightarrow u_i).$$

But this contradicts the fact that $bound_{Y_0}(\psi) \not\leq (t_j \Rightarrow u_i)$, which was established above.

□

Corollary 3.12. *Let $\Gamma \cup \{\varphi\} \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. Γ $\mathsf{Trans}_d^{\mathcal{H}}$-entails φ iff φ is $\mathsf{Trans}_d^{\mathcal{H}}$-entailed by some finite Γ_0 which is a subset of Γ.*

Proof. Similar to the proof of Corollary 3.9, only replacing $\mathsf{Refl}_d^{\mathcal{H}}$ and $\mathcal{C}\mathbf{T}_d^{\mathcal{H}}$ with $\mathsf{Trans}_d^{\mathcal{H}}$ and $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ respectively. □

4 Prefixed Tableaux

Intuitively, the modal rules considered so far suggest viewing the formulas in a branch as describing the valuation of a hypothetical model at a specific world. And in a sense, the # operations in the modal rules correspond to a change in world, and have induced us to forget much of the information regarding the previous world. This 'destructiveness' makes deriving a decision procedure from our rules difficult, and what's more, devising a system that is sound and complete with respect to the symmetric frames is impossible.
In this section we will be studying non-destructive prefixed tableau systems. Referring back to Section 2.3, this means that we take the object language to be $U = pSBI$ – the set of all prefixed signed bounding implications. To get an idea for the difficulty involved in systematically constructing closed tableaux in our unprefixed systems, consider a $\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau containing an open branch S. Further, assume both $\alpha = F(\Diamond\varphi \supset \underline{a}) \in S$ and $\beta = F(\Diamond\psi \supset \underline{b}) \in S$, where $\alpha \not\equiv \beta$. Then, in an attempt to close the tableau, we

may apply $F\Diamond$ to S around principle element α. Then, the # operation deletes β. On the other hand, we may apply $F\Diamond$ to S around principle element β, in which case the # operation deletes α. Clearly, the two different choices could lead to radically different $\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableaux down the line and one of them may close while the other doesn't. In general, there's no way to *a priori*, deterministically pick some sequence of rule applications that is guaranteed to lead to a closed tableau, if one exists. So, any deterministic decision procedure we could hope to define based on these unprefixed tableaux will require much bookkeeping and backtracking. Bookkeeping analogous to that which one needs to do when naively translating a non-deterministic algorithm into a deterministic one. Prefixes can take care of this bookkeeping quite naturally and will ensure we never have to backtrack. They do so by keeping track of all the worlds, past and present. For a prefix (w, σ), we think of:

- $w \in \Sigma$ as denoting a world in an $\mathcal{H}$-frame. We call w a *world label*. For the sake of presenting concrete examples, let us assume that the symbols $\mathtt{A}, \mathtt{B}, \mathtt{C}$ are world labels.
- $(w, v, \underline{t}) \in \sigma \subseteq \Sigma \times \Sigma \times \underline{H}$ as saying that the world denoted by v is accessible from the world denoted by w to degree t. We call $(w, v, \underline{t})$ a *constraint*.

With prefixes in hand, a branch of a tableau can describe an entire hypothetical model – not just the valuation at a specific state.
We shall use the following convenient notation: for $\beta \in SBI$, $p((w,\sigma)\beta) \coloneqq (w,\sigma)$; $sf((w,\sigma)\beta) \coloneqq \beta$; $world((w,\sigma)\beta) \coloneqq w$; $con((w,\sigma)\beta) \coloneqq \sigma$; and for a given set $X \subseteq pSBI$,

$$cons(X) \coloneqq \bigcup_{x \in X} con(x)$$

and

$$worlds(X) \coloneqq \{world(x) \mid x \in X\}.$$

Example 4.1. In Example 2.55 we considered the instantiation of a specific tableau rule in an unprefixed system. Let us now do the same for a rule in a prefixed system. Again, suppose $\mathcal{H} = \mathcal{H}^5$ as presented in Example 2.23. Consider the following tableau rule pEx.

$$(p\text{Ex}) \; \frac{X; (w,\sigma)F(\varphi \supset \underline{a})}{\mathcal{N}; (v, \sigma' \cup \{(w, v', \underline{t \wedge a})\})T(\varphi \supset \underline{t})}$$

Where v is any member of Σ and t any $\mathcal{H}$-truth value s.t. $(w, v, \underline{t}) \in \sigma'$. And v' is any symbol of Σ that is not in $worlds(\mathcal{N})$.

In the denominators and side condition of the rule, the $\mathcal{N}$ stands as an abreviation for the entire numerator, and σ' in turn is used as shorthand for $cons(\mathcal{N})$. The rule uses the usual metasymbols, along with ones for elements of a prefix: roman letters w, v for world labels; σ for constraints.
Suppose we uniformly substitute X with $\{(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{r})\})T(\underline{s} \supset \underline{1})\}$; w with $\mathtt{A}$; σ with $\varnothing$; φ with p; a with s; t with r; v with $\mathtt{B}$; and v' with $\mathtt{C}$. Then the set $\{(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{r})\})T(\underline{s} \supset \underline{1}), (\mathtt{A}, \varnothing)F(p \supset \underline{s})\}$ is an instantiation of the numerator, with principal element $(\mathtt{A}, \varnothing)F(p \supset \underline{s})$.
$\{(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{r})\})T(\underline{s} \supset \underline{1}), (\mathtt{A}, \varnothing)F(p \supset \underline{s}), (\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{r}), (\mathtt{A}, \mathtt{C}, \underline{r})\})T(p \supset \underline{r})\}$ is a corresponding instantiation of the denominator, with principal element $(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{r}), (\mathtt{A}, \mathtt{C}, \underline{r})\})T(p \supset \underline{r})$.

Definition 4.2. Let $\rho\mathcal{C}$ be some prefixed tableau system. The set of $p\mathcal{C}$-tableaux are defined as in Definition 2.57, with $U = pSBI$.
Definition 2.60 also still applies, but the introduction of prefixes requires the following slight adjustments. We shall say that a set of prefixed signed bounding implications is closed iff it contains $(w, \varnothing)\bot$ for some $w \in \Sigma$. And, a formula φ is $p\mathcal{C}$-derivable from a set of formulas Γ iff for some $w \in \Sigma$ and finite $\Gamma_0 \subseteq \Gamma$, there exists a closed $p\mathcal{C}$-tableau for $\{(w, \varnothing)T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma_0\} \cup \{(w, \varnothing)F(\underline{1} \supset \varphi)\}$

Now, the notions of soundness and completeness for prefixed tableau systems are as in Definition 2.61. Note, though, that we use a slightly different notational convention when naming prefixed systems: We shall use '$p\mathcal{C}\mathbf{X}$' to name the prefixed tableau system for the theory $\mathbf{X}$,
We proceed to study a prefixed tableau system for $\mathbf{K}^{\mathcal{H}}$.

$$\begin{aligned} p\mathcal{C}\mathbf{K}^{\mathcal{H}} :=& \{p\bot_1, p\bot_2, p\bot_3, p\bot_4, p\bot_5, pF{\geq}, pT{\geq}, pF{\leq}, pT{\leq}, pT{\wedge}, pF{\wedge}, pT{\vee}, pF{\vee}, \\ & pT{\supset}, pF{\supset}, p\mathbf{K}T\Box, p\mathbf{K}T\Diamond, p\mathbf{K}F\Box, p\mathbf{K}F\Diamond\} \end{aligned}$$

where these rules are defined below. Note that in all the rules, the entire numerator of the rule, denoted by $\mathcal{N}$, is carried to the denominator(s) of the rule. Furthermore, σ' is an abreviation for $cons(\mathcal{N})$

$$(p\bot_1)\ \frac{X; (w,\sigma)T(\underline{a} \supset \underline{b})}{\mathcal{N}; (w,\varnothing)\bot} \qquad \text{Where } a \nleq b$$

$$(p\bot_2)\ \frac{X; (w,\sigma)F(\underline{a} \supset \underline{b})}{\mathcal{N}; (w,\varnothing)\bot} \qquad \text{Where } a \leq b$$

$$(p\bot_3)\ \frac{X; (w,\sigma)F(\underline{0} \supset \varphi)}{\mathcal{N}; (w,\varnothing)\bot}$$

$$(p\bot_4)\ \frac{X; (w,\sigma)F(\varphi \supset \underline{1})}{\mathcal{N}; (w,\varnothing)\bot}$$

$$(p\bot_5)\ \frac{X; (w,\sigma)T(\underline{a} \supset \varphi); (w,\sigma')T(\varphi \supset \underline{b})}{\mathcal{N}; (w,\varnothing)\bot} \qquad \text{Where } a \nleq b$$

$$(pF{\geq})\ \frac{X; (w,\sigma)F(\underline{a} \supset \varphi)}{\begin{array}{l|c|l} \mathcal{N}; & \ldots & \mathcal{N}; \\ (w,\sigma')T(\varphi \supset \underline{t_1}) & & (w,\sigma')T(\varphi \supset \underline{t_n}) \end{array}}$$

Where $t_1, \ldots, t_n$ are all the maximal $\mathcal{H}$-truth values not above a, and $a \neq 0$.

$$(pT{\geq})\ \frac{X; (w,\sigma)T(\underline{a} \supset \varphi)}{\mathcal{N}; (w,\sigma')F(\varphi \supset \underline{t_i})}$$

Where t_i is any maximal $\mathcal{H}$-truth value not above a, and $a \neq 0$.

$$(pF{\leq})\ \frac{X; (w,\sigma)F(\varphi \supset \underline{a})}{\begin{array}{l|c|l} \mathcal{N}; & \ldots & \mathcal{N}; \\ (w,\sigma')T(\underline{u_1} \supset \varphi) & & (w,\sigma')T(\underline{u_k} \supset \varphi) \end{array}}$$

Where $u_1, \ldots, u_k$ are all the minimal $\mathcal{H}$-truth values not below a, and $a \neq 1$.

$$(pT{\leq})\ \frac{X; (w,\sigma)T(\varphi \supset \underline{a})}{\mathcal{N}; (w,\sigma')F(\underline{u_i} \supset \varphi)}$$

Where u_i is any minimal $\mathcal{H}$-truth value not below a, and $a \neq 1$.

$$(pT\wedge)\ \frac{X; (w,\sigma)T(\underline{a} \supset (\varphi \wedge \psi))}{\mathcal{N}; (w,\sigma')T(\underline{a} \supset \varphi); (w,\sigma')T(\underline{a} \supset \psi)}$$

Where $a \neq 0$.

$$(pF\wedge)\ \frac{X; (w,\sigma)F(\underline{a} \supset (\varphi \wedge \psi))}{\begin{array}{c|c} \mathcal{N}; (w,\sigma')F(\underline{a} \supset \varphi) & \mathcal{N}; (w,\sigma')F(\underline{a} \supset \psi) \end{array}}$$

Where $a \neq 0$.

$$(pT\vee)\ \frac{X; (w,\sigma)T((\varphi \vee \psi) \supset \underline{a})}{\mathcal{N}; (w,\sigma')T(\varphi \supset \underline{a}); (w,\sigma')T(\psi \supset \underline{a})}$$

Where $a \neq 1$.

$$(pF\vee)\ \frac{X; (w,\sigma)F((\varphi \vee \psi) \supset \underline{a})}{\begin{array}{c|c} \mathcal{N}; (w,\sigma')F(\varphi \supset \underline{a}) & \mathcal{N}; (w,\sigma')F(\psi \supset \underline{a}) \end{array}}$$

Where $a \neq 1$.

$$(pF{\supset})\ \frac{X; (w,\sigma)F(\underline{a} \supset (\varphi \supset \psi))}{\begin{array}{l|c|l} \mathcal{N}; & \dots & \mathcal{N}; \\ (w,\sigma')T(\underline{t_1} \supset \varphi); & & (w,\sigma')T(\underline{t_n} \supset \varphi); \\ (w,\sigma')F(\underline{t_1} \supset \psi) & & (w,\sigma')F(\underline{t_n} \supset \psi) \end{array}}$$

Where $t_1, \dots, t_n$ are all the $\mathcal{H}$-truth values below a except 0.

$$(pT{\supset})\ \frac{X; (w,\sigma)T(\underline{a} \supset (\varphi \supset \psi))}{\begin{array}{l|l} \mathcal{N}; & \mathcal{N}; \\ (w,\sigma')F(\underline{t_i} \supset \varphi) & (w,\sigma')T(\underline{t_i} \supset \psi) \end{array}}$$

Where t_i is any $\mathcal{H}$-truth value below a except 0.

$$(p\mathbf{K}T\Box)\ \frac{X; (w,\sigma)T(\underline{a} \supset \Box\varphi)}{\mathcal{N}; (v,\sigma')T(\underline{a \wedge t} \supset \varphi)}$$

Where v is any member of Σ and t any $\mathcal{H}$-truth value s.t. $(w, v, \underline{t}) \in \sigma'$.

$$(p\mathbf{K}T\Diamond)\ \frac{X; (w,\sigma)T(\Diamond\varphi \supset \underline{a})}{\mathcal{N}; (v,\sigma')T(\varphi \supset \underline{t \Rightarrow a})}$$

Where v is any member of Σ and t any $\mathcal{H}$-truth value s.t. $(w, v, \underline{t}) \in \sigma'$.

$$(p\mathbf{K}F\Box)\ \frac{X; (w,\sigma)F(\underline{a} \supset \Box\varphi)}{\begin{array}{l|c|l} \mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_1})\}) & \dots & \mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_n})\}) \\ F(\underline{a \wedge t_1} \supset \varphi) & & F(\underline{a \wedge t_n} \supset \varphi) \end{array}}$$

Where v is any symbol of Σ that is not in $worlds(\mathcal{N})$, and $t_1, \dots, t_n$ are all the $\mathcal{H}$-truth values s.t. $a \wedge t_i \neq 0$.

$$(p\mathbf{K}F\Diamond)\ \frac{X;(w,\sigma)F(\Diamond\varphi \supset \underline{a})}{\begin{array}{l|c|l} \mathcal{N};(v,\sigma' \cup \{(w,v,\underline{t_1})\}) & \dots & \mathcal{N};(v,\sigma' \cup \{(w,v,\underline{t_n})\}) \\ F(\varphi \supset \underline{t_1 \Rightarrow a}) & & F(\varphi \supset \underline{t_n \Rightarrow a}) \end{array}}$$

Where v is any symbol of Σ that is not in $worlds(\mathcal{N})$, and $t_1, \dots, t_n$ are all the $\mathcal{H}$-truth values s.t. $t_i \Rightarrow a \neq 1$.

Remark 4.3. In all the above rules, the constraints introduced in the denominators extend $\sigma' = cons(\mathcal{N})$. We could just as well instead extend the σ of the numerator. However, the current approach is chosen as it makes the later termination result (Lemma 4.20) easier to prove.

As apposed to our unprefixed systems, none of the these rules require us to discard elements in a branch. So, any tableau we deal with in this section can be effectively represented by a labeled tree (Recall Definition 2.74).

Definition 4.4. Let $(\mathcal{T}, U)$ be a labelled tree, and suppose $\{\mathcal{B}^i\}_{i\in I}$ are all of the branches of $\mathcal{T}$. The tableau ***corresponding*** to $(\mathcal{T}, U)$ (denoted $T_{(\mathcal{T},U)}$) is simply the collection $\{U(\mathcal{B}^i)\}_{i\in I}$.
We will say that a branch $\mathcal{B}$ of $\mathcal{T}$ is closed iff $U(\mathcal{B})$ is closed. Otherwise, we say that $\mathcal{B}$ is open.
We will say that $(\mathcal{T}, U)$ is closed iff all the branches of $\mathcal{T}$ are closed; otherwise, we say it is open.

Remark 4.5. For an arbitrary labelled tree $(\mathcal{T}, U)$, $T_{(\mathcal{T},U)}$ is not necessarily a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau, in the strict sense of Definition 2.57. However, the labeled trees that will crop up in our examples and decision procedure will have the property that $T_{(\mathcal{T},U)}$ is in fact a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{U(r)\}$, where r is the root node of $\mathcal{T}$ (see Lemma 4.22).

In addition to 'closed' and 'open', It will be useful to apply other tableau terminology to labelled trees. For instance, let us introduce the notion of applying tableau rules to labelled trees. Essentially, the following definition allows us to talk about 'applying a rule ρ to labelled tree $(\mathcal{T}, U)$' as a shorthand for actually saying that we extend $(\mathcal{T}, U)$ such that the corresponding tableau is derivable via an application of ρ to $T_{(\mathcal{T},U)}$.

Definition 4.6. Suppose we have a labelled tree $(\mathcal{T}, U)$, and that $T_{(\mathcal{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau. Let $\rho \in p\mathcal{C}\mathbf{K}^{\mathcal{H}}$, and suppose $T^*_{(\mathcal{T},U)}$ is some $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau derived from $T_{(\mathcal{T},U)}$ via an application of ρ. Then, any labeled tree $(\mathcal{T}^*, U^*)$ extending $(\mathcal{T}, U)$ for which $T_{(\mathcal{T}^*,U^*)} = T^*_{(\mathcal{T},U)}$, can be said to have been derived via an **application** of ρ to $(\mathcal{T}, U)$. Further, If $\mathcal{B}$ is a branch of $\mathcal{T}$ but not of $\mathcal{T}^*$, we say that ρ was applied to branch $\mathcal{B}$.

Example 4.7. For this example, suppose $\mathcal{H} = \mathcal{H}^3$ and let $\varphi \equiv \neg\Diamond\neg p \supset \Box p$. Figure 4.1 illustrates a labeled tree whose corresponding tableau is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A}, \varnothing)F(\underline{1} \supset \varphi)\}$. Let us walk through the rule applications that were performed to derive the tableau.

- We start with the labeled tree $T_{(\mathcal{T},U)}$ consisting of only node 1, which is labeled with $(\mathtt{A}, \varnothing)F\big(\underline{1} \supset \varphi\big)$. Take the branch $\mathcal{B}$ of $\mathcal{T}$ consisting of node 1 (node 1 is the root and currently the only leaf node). We apply $pF\!\supset$ to $\mathcal{B}$ by adding nodes 2,3,4 and 5 and labeling them to get the new labelled tree $(\mathcal{T}^*, U^*)$. Note that $T_{(\mathcal{T},U)} = \{U(\mathcal{B})\} = \{\{(\mathtt{A}, \varnothing)F\big(\underline{1} \supset (((\Diamond p \supset \underline{0}) \supset \underline{0}) \supset \Box p)\big)\}\}$ is a base $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A}, \varnothing)F(\underline{1} \supset \varphi)\}$, and $U(\mathcal{B})$ is an instantiation of the numerator of $pF\!\supset$. Further,

$$① \ (\mathtt{A},\varnothing)F\big(\underline{1}\supset(((\Diamond p\supset\underline{0})\supset\underline{0})\supset\Box p)\big)$$

$$② \ (\mathtt{A},\varnothing)T\big(\underline{1}\supset((\Diamond p\supset\underline{0})\supset\underline{0})\big) \quad ④ \ (\mathtt{A},\varnothing)T\big(\underline{h}\supset((\Diamond p\supset\underline{0})\supset\underline{0})\big)$$

$$③ \ (\mathtt{A},\varnothing)F\big(\underline{1}\supset\Box p\big) \quad ⑤ \ (\mathtt{A},\varnothing)F\big(\underline{h}\supset\Box p\big)$$

$$⑥ \ (\mathtt{A},\varnothing)F\big(\underline{1}\supset\Diamond(p\supset\underline{0})\big) \quad ⑦ \ (\mathtt{A},\varnothing)T\big(\underline{1}\supset\underline{0}\big)$$

$$⑧ \ (\mathtt{A},\varnothing)F\big(\underline{h}\supset\Diamond(p\supset\underline{0})\big) \quad ⑨ \ (\mathtt{A},\varnothing)T\big(\underline{h}\supset\underline{0}\big)$$

$$⑩ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{h})\})F\big(h\supset p\big) \quad ⑪ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{1}\supset p\big)$$

$$⑫ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(p\supset\underline{h}\big)$$

$$⑬ \ (\mathtt{A},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\Diamond(p\supset\underline{0})\supset\underline{h}\big)$$

$$⑭ \ (\mathtt{A},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\Diamond(p\supset\underline{0})\supset\underline{0}\big)$$

$$⑮ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big((p\supset\underline{0})\supset\underline{h}\big)$$

$$⑯ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big((p\supset\underline{0})\supset\underline{0}\big)$$

$$⑰ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{1}\supset(p\supset\underline{0})\big)$$

$$⑱ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{h}\supset(p\supset\underline{0})\big)$$

$$⑲ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\underline{1}\supset p\big) \quad ㉑ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\underline{h}\supset p\big)$$

$$⑳ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{1}\supset\underline{0}\big) \quad ㉒ \ (\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{h}\supset\underline{0}\big)$$

Figure 4.1

$T_{(\mathscr{T}^*,U^*)} = \{\{(\mathtt{A},\varnothing)F\big(\underline{1} \supset (((\Diamond p \supset \underline{0}) \supset \underline{0}) \supset \Box p)\big), (\mathtt{A},\varnothing)T\big(\underline{1} \supset ((\Diamond p \supset \underline{0}) \supset \underline{0})\big), (\mathtt{A},\varnothing)F\big(\underline{1} \supset \Box p\big)\}, \{(\mathtt{A},\varnothing)F\big(\underline{1} \supset (((\Diamond p \supset \underline{0}) \supset \underline{0}) \supset \Box p)\big), (\mathtt{A},\varnothing)T\big(\underline{h} \supset ((\Diamond p \supset \underline{0}) \supset \underline{0})\big), (\mathtt{A},\varnothing)F\big(\underline{h} \supset \Box p\big)\}\}$ is derivable from $T_{(\mathscr{T},U)}$ via an application of $pF{\supset}$ to $U(\mathscr{B})$. This is what justifies our saying that we apply ρ to $\mathscr{B}$ of $(\mathscr{T}, U)$ to get $(\mathscr{T}^*, U^*)$. Also, observe that since there were multiple corresponding denominators, we split/forked the branch $\mathscr{B}$.

- Consider the branch $\mathscr{B}^*$ of $\mathscr{T}^*$ with leaf node 3. We add nodes 6 and 7 with their labels to get a new labelled tree. This reflects an application of $pT{\supset}$ to $\mathscr{B}^*$ around principle element $(\mathtt{A},\varnothing)T\big(\underline{1} \supset ((\Diamond p \supset \underline{0}) \supset \underline{0})\big)$ (the label of node 2), with $t_i = 1$ in the side condition.
- Consider the branch of the new tree with leaf node 6. Again apply $pT{\supset}$ around principle element $(\mathtt{A},\varnothing)T\big(\underline{1} \supset ((\Diamond p \supset \underline{0}) \supset \underline{0})\big)$ but now with $t_i = h$ in the side condition. This amounts to adding nodes 8 and 9.
- Consider the branch of the new tree with leaf node 8. Apply $p\mathbf{K}F\Box$ to this branch around principle element $(\mathtt{A},\varnothing)F\big(\underline{1} \supset \Box p\big)$. This amounts to adding nodes 10 and 11.
- Consider the branch of the new tree with leaf node 11. Apply $pF{\geq}$ to this branch around principle element $(\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{1} \supset p\big)$. This amounts to adding node 12.
- Consider the branch of the new tree with leaf node 12. Apply $pF{\geq}$ to this branch around principle element $(\mathtt{A},\varnothing)F\big(\underline{1} \supset \Diamond(p \supset \underline{0})\big)$. This amounts to adding node 13.
- Adding 14 reflects an application of $F{\geq}$ around principle element $(\mathtt{A},\varnothing)F\big(\underline{h} \supset \Diamond(p \supset \underline{0})\big)$.
- Consider the branch of the new tree with leaf node 14. Apply $p\mathbf{K}T\Diamond$ to this branch around principle element $(\mathtt{A},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\Diamond(p \supset \underline{0}) \supset \underline{h}\big)$. This amounts to adding node 15.
- Consider the branch of the new tree with leaf node 15. Apply $p\mathbf{K}T\Diamond$ to this branch around principle element $(\mathtt{A},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big(\Diamond(p \supset \underline{0}) \supset \underline{0}\big)$. This amounts to adding node 16.
- Consider the branch of the new tree with leaf node 16. Apply $pT{\leq}$ to this branch around principle element $(\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big((p \supset \underline{0}) \supset \underline{h}\big)$. This amounts to adding node 17.
- Consider the branch of the new tree with leaf node 17. Apply $pT{\leq}$ to this branch around principle element $(\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})T\big((p \supset \underline{0}) \supset \underline{0}\big)$. This amounts to adding node 18.
- Consider the branch of the new tree with leaf node 18. Apply $pF{\supset}$ to this branch around principle element $(\mathtt{B},\{(\mathtt{A},\mathtt{B},\underline{1})\})F\big(\underline{1} \supset (p \supset \underline{0})\big)$. This amounts to adding nodes 19, 20, 21 and 22.

We have applied a finite sequence of rules to labeled trees, starting with a labelled tree whose corresponding tableau is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A},\varnothing)F(\underline{1} \supset \varphi)\}$. So, what we have ended up with in Figure 4.1 is in fact a labeled tree whose corresponding tableau is also a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A},\varnothing)F(\underline{1} \supset \varphi)\}$. As the reader can convince themselves, no further sequence of rule applications will close the branch with leaf node 22. What's more, in a sense this branch is in a completed state. More precisely, the branch is in the special state of being 'downward saturated' (which we define properly in Definition 4.13). By proving soundness we will show that if φ has a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-proof, then it is valid. However, downward saturation implies that no $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-proof for φ exists, since (as we will prove generally in Section 4.2), the prefixes and atomic bounding implications in a downward saturated branch of a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A},\varnothing)F(\underline{1} \supset \varphi)\}$ provide all the information needed to construct a counter $\mathcal{H}$-model for φ. Consider the branch $\mathscr{B}$ with leaf node 22. Let us get a taste for the proof of Lemma 4.15 by allowing the labels in $\mathscr{B}$ to guide the construction of an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$. Take W to be $worlds(U(\mathscr{B})) = \{\mathtt{A}, \mathtt{B}\}$. Nodes 11 and 21 tell us to set $V(\mathtt{B}, p) \coloneqq h$. Nothing as explicit is implied by $U(\mathscr{B})$ about $V(\mathtt{A}, p)$. So let us arbitrarily set $V(\mathtt{A}, p) \coloneqq 0$. Let us define R according to $cons(U(\mathscr{B})) = \{(\mathtt{A}, \mathtt{B}, \underline{1})\}$. This leads us to set $R \coloneqq \{((\mathtt{A}, \mathtt{A}), 0), ((\mathtt{A}, \mathtt{B}), 1), ((\mathtt{B}, \mathtt{A}), 0), ((\mathtt{B}, \mathtt{B}), 0)\}$.

As it turns out, $\mathfrak{M}$ is the same as the model given in Example 2.42, with $\mathfrak{s} = \mathtt{A}$ and $\mathfrak{v} = \mathtt{B}$. And we showed there that $\mathfrak{M}$ is a counter model for φ.
Now, one may object that our choice for the value of $V(\mathtt{A}, p)$ above was not so arbitrary. We leave it to these skeptics to confirm that indeed any choice for the $\mathcal{H}^3$-truth value of $V(\mathtt{A}, p)$ would still leave us with a counter model for φ.

4.1 Soundness

We shall now prove $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ is sound with respect to the class of all $\mathcal{H}$-frames. Before doing so, let us define the notion of *satisfiability* for elements of $\mathcal{P}(pSBI)$ (recall that 2.48 only applies to sets of unprefixed signed formulas). In doing so, we make precise the intuition that a symbol from Σ occurring in some prefix is a name for a possible world.

Definition 4.8. Let S be a set of prefixed signed bounding implications and let $\mathfrak{M} = ((W, R), V)$ be an $\mathcal{H}$-model.
An ***interpretation*** of S in $\mathfrak{M}$ is any map $I : worlds(S) \to W$ s.t. if $(w, v, \underline{t}) \in cons(S)$, then I is defined for w and v (i.e. $w, v \in worlds(S)$) and $R(I(w), I(v)) = t$.
We say S is ***satisfied under*** I if for each $(w, \sigma)\beta \in S$, it is the case that β is satisfied by $\mathfrak{M}$ at $I(w)$.
Further, let $\mathcal{F}$ be a class of $\mathcal{H}$-frames. We say S is ***$\mathcal{F}$-satisfiable*** iff there exists an $\mathcal{H}$-model $\mathfrak{M}$ based on a frame from $\mathcal{F}$, and an interpretation I of S in $\mathfrak{M}$ s.t. S is satisfied under I.
In the case where $\mathcal{F}$ is the class of all $\mathcal{H}$-frames, we simply say that S is satisfiable.

Now, Definition 2.62 makes sense in the context of prefixed systems. The following Lemma is analogues to Lemma 3.2.

Lemma 4.9. *Let $\mathcal{F}$ be an arbitrary class of $\mathcal{H}$-frames. ρ preserves $\mathcal{F}$-satisfiability for each $\rho \in p\mathcal{C}\boldsymbol{K}^{\mathcal{H}}$.*

Proof. It suffices to show that for each such rule, if (an instantiation of) the numerator $\mathcal{N}$ is $\mathcal{F}$-satisfiable, then (the corresponding instantiation of) at least one of the denominators $\mathcal{D}$ is $\mathcal{F}$-satisfiable.
Let $\rho \in p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ and suppose that the numerator $\mathcal{N}$ of ρ is $\mathcal{F}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$ based on a frame from $\mathcal{F}$, and an interpretation I of $\mathcal{N}$ in $\mathfrak{M}$ s.t. $\mathcal{N}$ is satisfied under I.
We now need to consider each case individually. For the non-modal rules, the argument remains essentially the same as for the unprefixed systems (Lemma 3.2). For illustrative purposes, we shall consider one such rule.

- $\rho = pT\supset$.

$$\mathcal{N} = X; (w, \sigma)T(\underline{a} \supset (\varphi \supset \psi))$$

So $T(\underline{a} \supset (\varphi \supset \psi))$ is satisfied by $\mathfrak{M}$ at $I(w)$. That is, $V(I(w), \underline{a} \supset (\varphi \supset \psi)) = 1$. Or equivalently,

$$\begin{aligned} a &\leq V(I(w), \varphi \supset \psi) \\ &= V(I(w), \varphi) \Rightarrow V(I(w), \psi) \end{aligned}$$

and so $a \wedge V(I(w), \varphi) \leq V(I(w), \psi)$.
Let t_i be any $\mathcal{H}$-truth value below a except 0. Assuming $t_i \leq V(I(w), \varphi)$, then $t_i \leq a \wedge V(I(w), \varphi) \leq V(I(w), \psi)$.
Thus, we have either $t_i \nleq V(I(w), \varphi)$ (so $(w, \sigma')F(\underline{t_i} \supset \varphi)$ is satisfied by $\mathfrak{M}$ at $I(w)$) or $t_i \leq V(I(w), \varphi)$, in which case $t_i \leq V(I(w), \psi)$ (so $(w, \sigma')T(\underline{t_i} \supset \psi)$ is satisfied by $\mathfrak{M}$ at $I(w)$).
Therefore, for any t_i below a except 0, at least one of the denominators is satisfied under I.

We now present the proof for the modal rules.

- $\rho = p\mathbf{K}T\Box$.

$$\mathcal{N} = X; (w, \sigma)T(\underline{a} \supset \Box\varphi)$$

Let v be any member of Σ and t any $\mathcal{H}$-truth value s.t. $(w, v, \underline{t}) \in \sigma' = \mathit{cons}(\mathcal{N})$. By the definition of an interpretation, we must have that $w, v \in \mathit{worlds}(\mathcal{N})$ and $R(I(w), I(v)) = t$.
Since $(w, \sigma)T(\underline{a} \supset \Box\varphi) \in \mathcal{N}$ and $\mathcal{N}$ is satisfied under I, we have

$$\begin{aligned} a &\leq V(I(w), \Box\varphi) \\ &= \bigwedge\{R(I(w), \mathfrak{s}) \Rightarrow V(\mathfrak{s}, \varphi) \mid \mathfrak{s} \in W\} \\ &\leq R(I(w), I(v)) \Rightarrow V(I(v), \varphi) && (I(v) \in W) \end{aligned}$$

Or equivalently, $a \wedge t \leq V(I(v), \varphi)$. Thus, $(v, \mathit{cons}(\mathcal{N}))T(\underline{a \wedge t} \supset \varphi)$ is satisfied by $\mathfrak{M}$ at $I(v)$

- $\rho = p\mathbf{K}T\Diamond$.

$$\mathcal{N} = X; (w, \sigma)T(\Diamond\varphi \supset \underline{a})$$

Let v be any member of Σ and t any $\mathcal{H}$-truth value s.t. $(w, v, \underline{t}) \in \mathit{cons}(\mathcal{N})$. By the definition of an interpretation, we must have that $w, v \in \mathit{worlds}(\mathcal{N})$ and $R(I(w), I(v)) = t$.
Since $(w, \sigma)T(\Diamond\varphi \supset \underline{a}) \in \mathcal{N}$ and $\mathcal{N}$ is satisfied under I, we have

$$\begin{aligned} a &\geq V(I(w), \Diamond\varphi) \\ &= \bigvee\{R(I(w), \mathfrak{s}) \wedge V(\mathfrak{s}, \varphi) \mid \mathfrak{s} \in W\} \\ &\geq R(I(w), I(v)) \wedge V(I(v), \varphi) && (I(v) \in W) \end{aligned}$$

Or equivalently, $V(I(v), \varphi) \leq t \Rightarrow a$. Thus, $(v, \mathit{cons}(\mathcal{N}))T(\varphi \supset \underline{t \Rightarrow a})$ is satisfied by $\mathfrak{M}$ at $I(v)$.

- $\rho = p\mathbf{K}F\Box$.

$$\mathcal{N} = X; (w, \sigma)F(\underline{a} \supset \Box\varphi)$$

So $F(\underline{a} \supset \Box\varphi)$ is satisfied by $\mathfrak{M}$ at $I(w)$. That is, $V(I(w), \underline{a} \supset \Box\varphi) \neq 1$. Or equivalently,

$$\begin{aligned} a &\nleq V(I(w), \Box\varphi) \\ &= \bigwedge\{R(I(w), \mathfrak{s}) \Rightarrow V(\mathfrak{s}, \varphi) \mid \mathfrak{s} \in W\}. \end{aligned}$$

Thus, for some $\mathfrak{s} \in W$, we have $a \nleq R(I(w), \mathfrak{s}) \Rightarrow V(\mathfrak{s}, \varphi)$. Or equivalently,

$$a \wedge R(I(w), \mathfrak{s}) \nleq V(\mathfrak{s}, \varphi). \tag{4.9.1}$$

Suppose $R(I(w), \mathfrak{s}) = t_i \in H$. Clearly (4.9.1) implies $a \wedge t_i \neq 0$. Let $v \in \Sigma$ be any symbol that is not already in $worlds(\mathcal{N})$. We extend the interpretation I to v. Specifically, consider $I' \coloneqq I \cup \{(v, \mathfrak{s})\}$, which is an interpretation of $\mathcal{D} = \mathcal{N}; (v, cons(\mathcal{N}) \cup \{(w, v, \underline{t_i})\})F(\underline{a \wedge t_i} \supset \varphi)$ in $\mathfrak{M}$.
By (4.9.1), we have that $\mathcal{D}$ is satisfied under I'.

- $\rho = p\mathbf{K}F\Diamond$.

$$\mathcal{N} = X; (w, \sigma)F(\Diamond\varphi \supset \underline{a})$$

So $F(\Diamond\varphi \supset \underline{a})$ is satisfied by $\mathfrak{M}$ at $I(w)$. That is, $V(I(w), \Diamond\varphi \supset \underline{a}) \neq 1$. Or equivalently,

$$\begin{aligned} a &\ngeq V(I(w), \Diamond\varphi) \\ &= \bigvee\{R(I(w), \mathfrak{s}) \wedge V(\mathfrak{s}, \varphi) \mid \mathfrak{s} \in W\}. \end{aligned}$$

Thus, for some $\mathfrak{s} \in W$, we have $R(I(w), \mathfrak{s}) \wedge V(\mathfrak{s}, \varphi) \nleq a$. Or equivalently,

$$V(\mathfrak{s}, \varphi) \nleq R(I(w), \mathfrak{s}) \Rightarrow a. \tag{4.9.2}$$

Suppose $R(I(w), \mathfrak{s}) = t_i \in H$. Clearly (4.9.2) implies $t_i \Rightarrow a \neq 1$. Let $v \in \Sigma$ be any symbol that is not already in $worlds(\mathcal{N})$. We extend the interpretation I to v. Specifically, consider $I' \coloneqq I \cup \{(v, \mathfrak{s})\}$, which is an interpretation of $\mathcal{D} = \mathcal{N}; (v, cons(\mathcal{N}) \cup \{(w, v, \underline{t_i})\})F(\varphi \supset \underline{t_i \Rightarrow a})$ in $\mathfrak{M}$.
By (4.9.2), we have that $\mathcal{D}$ is satisfied under I'.

□

Lemma 4.10. *If there exists a closed $p\mathcal{C}\boldsymbol{K}^{\mathcal{H}}$-tableau for a set of prefixed signed bounding implications X, then X is not satisfiable.*

Proof. We can use the exact same proof as that given for Lemma 2.63. However, now we must keep in mind that $U = pSBI$ and satisfiability is in terms of interpretations. □

Proposition 4.11. *$p\mathcal{C}\boldsymbol{K}^{\mathcal{H}}$ is sound with respect to the class of all $\mathcal{H}$-frames. I.e., for every formula φ, if $\vdash_{p\mathcal{C}\boldsymbol{K}^{\mathcal{H}}} \varphi$ then φ is valid.*

Proof. Let φ be an arbitrary formula and suppose that $\vdash_{p\mathcal{C}\mathbf{K}^{\mathcal{H}}} \varphi$. That is, for some $w \in \Sigma$, there exists a closed $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $X = \{(w, \varnothing)F(\underline{1} \supset \varphi)\}$. Then, by the above lemma, it means that X is not satisfiable. In other words, for every $\mathcal{H}$-model $\mathfrak{M}$ and interpretation I of X in $\mathfrak{M}$, it is the case that X is not satisfied under I.
For the sake of contradiction, assume φ is not valid. Then, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$ and world $\mathfrak{s} \in W$ s.t. $V(\mathfrak{s}, \varphi) \neq 1$ (or equivalently, $V(\mathfrak{s}, \underline{1} \supset \varphi) \neq 1$). Consider the interpretation $I = \{(w, \mathfrak{s})\}$ of X in $\mathfrak{M}$. Clearly X is satisfied under I, contradicting that X is not satisfiable. Thus, we must have that φ is in fact valid. □

4.2 Completeness

We may now approach proving completeness in much the same way as we did in section 3. That is, we could define the abstract notion of a maximal-consistent set of prefixed formulas and use such sets to

construct a (possibly infinite) canonical model [9]. Rather, in this section we do something that was not easily achieved in the previous section. We describe a decision procedure that, given a formula φ, must produce a tableau proof for φ if one exists and, if one does not, will give us the information necessary to construct a counter model for φ. This is a more constructive approach to proving completeness, one more in line with the spirit of the actual proof systems we are studying. Further, it will allow us to prove a finite frame property.
Along with helping prove these metatheorems, the decision procedure is also useful in its own right. It provides an algorithm that can be implemented on a computer and used to automate the (often arduous) process of establishing validity or finding a counter-model of a many-valued modal formula.

We use a labeled tree as the main data structure in the decision procedure for deriving a *desired* tableau. As just mentioned, a desired tableau for a non-valid formula is one that provides enough information to construct a counter model. This rough idea of 'enough information' is captured by the notion of **downward saturation**, which was introduced informally in Example 4.7.

Definition 4.12. Let S be any set of prefixed signed bounding implications. We define the relation

$$R_S \coloneqq \{((w, v), t) \in \Sigma^2 \times H \mid (w, v, \underline{t}) \in cons(S)\}$$

Definition 4.13. Let S be any set of prefixed signed bounding implications. S is said to be ***downward saturated*** iff all of the following conditions hold:

1. If $(w, v, \underline{t}) \in cons(S)$ for some $w, v \in \Sigma$, $t \in H$, then $w, v \in worlds(S)$. Further, R_S is a partial function from $worlds(S)^2$ to H.

2. For each rule $\rho \in \{p\bot_1, p\bot_2, p\bot_3, p\bot_4, p\bot_5\}$, S is not an instantiation of the numerator of ρ.

3. If $(w, \sigma)T(\underline{a} \supset (\varphi \wedge \psi)) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 0$, then we have $(w, \sigma')T(\underline{a} \supset \varphi) \in S$ and $(w, \sigma')T(\underline{a} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

4. If $(w, \sigma)F(\underline{a} \supset (\varphi \wedge \psi)) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 0$, then we have $(w, \sigma')F(\underline{a} \supset \varphi) \in S$ or $(w, \sigma')F(\underline{a} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

5. If $(w, \sigma)T((\varphi \vee \psi) \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 1$, then we have $(w, \sigma')T(\underline{a} \supset \varphi) \in S$ and $(w, \sigma')T(\psi \supset \underline{a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

6. If $(w, \sigma)F((\varphi \vee \psi) \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 1$, then we have $(w, \sigma')F(\varphi \supset \underline{a}) \in S$ or $(w, \sigma')F(\psi \supset \underline{a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

7. If $(w, \sigma)F(\underline{a} \supset (\varphi \supset \psi)) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then for some $t_i \in H$ s.t. $t_i \leq a$ and $t_i \neq 0$, we have $(w, \sigma')T(\underline{t_i} \supset \varphi) \in S$ and $(w, \sigma')F(\underline{t_i} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

8. If $(w, \sigma)T(\underline{a} \supset (\varphi \supset \psi)) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then for all $t_i \in H$ s.t. $t_i \leq a$ and $t_i \neq 0$, we have $(w, \sigma')F(\underline{t_i} \supset \varphi) \in S$ or $(w, \sigma')T(\underline{t_i} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

9. If $(w, \sigma)T(\underline{a} \supset \Box\varphi) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then for all $v \in \Sigma$ and $t \in H$ s.t. $(w, v, \underline{t}) \in cons(S)$, we have $(v, \sigma')T(\underline{a \wedge t} \supset \varphi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

[9] See [26], in which this is done in the context of prefixed systems for standard modal logics.

10. If $(w,\sigma)T(\Diamond\varphi \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then for all $v \in \Sigma$ and $t \in H$ s.t. $(w, v, \underline{t}) \in cons(S)$, we have $(v, \sigma')T(\varphi \supset \underline{t \Rightarrow a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

11. If $(w,\sigma)F(\underline{a} \supset \Box\varphi) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then there exists some $v \in \Sigma$ and $t_i \in H$ s.t. $a \wedge t_i \neq 0$, $(w, v, \underline{t_i}) \in cons(S)$ and $(v, \sigma')F(\underline{a \wedge t_i} \supset \varphi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

12. If $(w,\sigma)F(\Diamond\varphi \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a, then there exists some $v \in \Sigma$ and $t_i \in H$ s.t. $t_i \Rightarrow a \neq 1$, $(w, v, \underline{t_i}) \in cons(S)$ and $(v, \sigma')F(\varphi \supset \underline{t_1 \Rightarrow a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

13. If $(w,\sigma)F(\underline{a} \supset \varphi) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 0$; and φ has one of the following forms

    - p (a propositional variable)
    - $\psi \vee \theta$
    - $\Diamond\psi$

    Then, for some t which is a maximal truth value not above a, $(w, \sigma')T(\varphi \supset \underline{t}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

14. If $(w,\sigma)F(\varphi \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 1$; and φ has one of the following forms

    - p (a propositional variable)
    - $\psi \wedge \theta$
    - $\psi \supset \theta$
    - $\Box\psi$

    Then, for some u which is a minimal truth value not below a, $(w, \sigma')T(\underline{u} \supset \varphi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

15. If $(w,\sigma)T(\underline{a} \supset \varphi) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a; and φ has one of the following forms

    - $\psi \vee \theta$
    - $\Diamond\psi$

    Then, for all $t \in H$ which are maximal truth values not above a, $(w, \sigma')F(\varphi \supset \underline{t}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

16. If $(w,\sigma)T(\varphi \supset \underline{a}) \in S$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a; and φ has one of the following forms

    - $\psi \wedge \theta$
    - $\psi \supset \theta$
    - $\Box\psi$

    Then, for all $u \in H$ which are minimal truth values not below a, $(w, \sigma')F(\underline{u} \supset \varphi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

Remark 4.14. In this definition, we will mainly be concerned with the case in which S is a branch of a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau. Then, Conditions (3) to (12) may be seen as asserting that the branch is closed under applications of the rules $pT\wedge, pF\wedge, pT\vee, pF\vee, pT\supset, pF\supset, p\mathbf{K}T\Box, p\mathbf{K}T\Diamond, p\mathbf{K}F\Box$ and $p\mathbf{K}F\Diamond$ respectively. Conditions (13) to (16) are in a sense restricted closure conditions for the reversal rules. Essentially, the restrictions reflect the fact that we will wish to block the indiscriminate application of reversal rules to branches so as to ensure the termination of a procedure that constructs tableaux (which we do in Section 4.2.1).

The usefulness of this definition is formally embodied in the following Lemma.

Lemma 4.15. *Let S be any set of prefixed signed bounding implications. If S is downward saturated, then S is satisfiable.*

Proof. Suppose S is a downward saturated set of prefixed signed bounding implications.
Define the $\mathcal{H}$-frame (W, R) where $W := worlds(S)$ and for all $w, v \in W$,

$$R(w,v) := \begin{cases} R_S(w,v) & \text{if } R_S(w,v) \text{ defined} \\ 0 & \text{otherwise} \end{cases}$$

It follows from Condition (1) of downward saturation that $R : W^2 \to H$ is a well-defined function.
Now, consider an $\mathcal{H}$-model $\mathfrak{M}_S = ((W, R), V)$ where V is any valuation s.t. for every $w \in W$ and propositional variable p,

$$\begin{aligned} &\bigvee\{a \in H \mid (w,\sigma)T(\underline{a} \supset \varphi) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\} \\ &\leq V(w,p) \\ &\leq \bigwedge\{b \in H \mid (w,\sigma)T(\varphi \supset \underline{b}) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\} \end{aligned}$$

Such a V must exist. For assume, on the contrary, that for some formula φ and $w \in \Sigma$,

$$\begin{aligned} &\bigvee\{a \in H \mid (w,\sigma)T(\underline{a} \supset \varphi) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\} \\ &\not\leq \bigwedge\{b \in H \mid (w,\sigma)T(\varphi \supset \underline{b}) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\}. \end{aligned}$$

Then, we must have some $(w,\sigma)T(\underline{a} \supset \varphi) \in S$ and $(w,\sigma')T(\varphi \supset \underline{b}) \in S$ where $a \not\leq b$.
But this implies that S is an instantiation of the numerator of $p\bot_5$, and so Condition (2) of downward saturation is violated. This contradicts the fact that S is downward saturated.
We call $\mathfrak{M}_S$ an $\mathcal{H}$-model ***induced by*** S (note that there may be multiple such models with distinct valuations).

We proceed to prove, by induction on the structure of formulas, that for every formula φ, $P(\varphi)$ holds. Where $P(\varphi)$ is the statement:

for all $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$, $a \in H$ and β that bound φ by a,
if $(w,\sigma)\beta \in S$, then β is satisfied by $\mathfrak{M}_S$ at w

For the base case, let $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$, $a \in H$ and consider the following cases where β is atomic.

- φ is of the form $\underline{b}$ for some $b \in H$.
  Suppose $(w, \sigma)\beta \in S$ where β bounds φ by a. Assume that β is not satisfied by $\mathfrak{M}_S$ at w. Then, we must have that S is an instantiation of the numerator of one of the rules $p\bot_1, p\bot_2, p\bot_3$ or $p\bot_4$. Hence, S cannot be downward saturated. But, by our original supposition, S is downward saturated. Thus, β must be satisfied by $\mathfrak{M}_S$ at w.

- φ is some propositional variable p.
  Let β be a signed bounding implication that bounds φ by a. So β is of the form $T(\underline{a} \supset p), T(p \supset \underline{a}), F(\underline{a} \supset p)$ or $F(p \supset \underline{a})$. We need to consider each case separately.

  - β is of the form $T(\underline{a} \supset p)$.
    Suppose $(w, \sigma)T(\underline{a} \supset p) \in S$. Then, by the definition of V, we have $a \leq V(w, p)$. I.e. $V(w, a \supset p) = 1$. Thus, β is satisfied by $\mathfrak{M}_S$ at w.
  - β is of the form $F(\underline{a} \supset p)$.
    Suppose $(w, \sigma)F(\underline{a} \supset p) \in S$. By Condition (2) of downward saturation, we must have $a \neq 0$. And so, by Condition (13) of downward saturation, we must have $(w, \sigma')T(p \supset \underline{t}) \in S$ for some $t \in max(\{c \in H \mid a \nleq c\})$ and $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Then, by the definition of V, we have $V(w, p) \leq t$. Thus, by the contrapositive of Lemma 2.11 (1), we must have $a \nleq V(w, p)$. I.e. $V(w, a \supset p) \neq 1$, and hence, β is satisfied by $\mathfrak{M}_S$ at w.

  The other two cases are treated similarly.

For the inductive case, let ψ, θ be arbitrary formulas and suppose $P(\psi)$, $P(\theta)$ hold (this is our induction hypothesis) and assume φ is constructed from ψ and/or θ. For the cases in which the principal connective of φ is not modal, showing $P(\varphi)$ holds by appealing to the induction hypothesis is quite routine. We shall only look at one such case in order to illustrate the general approach.

- φ is of the form $\psi \supset \theta$
  Let $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$, $a \in H$ and β a signed bounding implication that bounds φ by a. So β is of the form $T(\underline{a} \supset (\psi \supset \theta)), T((\psi \supset \theta) \supset \underline{a}), F(\underline{a} \supset (\psi \supset \theta))$ or $F((\psi \supset \theta) \supset \underline{a})$. We consider each case.

<u>Case 1.1</u> β is of the form $T(\underline{a} \supset (\psi \supset \theta))$.
Suppose $(w, \sigma)T(\underline{a} \supset (\psi \supset \theta)) \in S$ and consider $t_i := a \wedge V(w, \psi)$. If $t_i = 0$ then $T(\underline{a} \supset (\psi \supset \theta))$ is clearly satisfied at w. So assume $t_i \neq 0$. Since we also have that $t_i \leq a$, Condition (8) implies that $(w, \sigma')F(\underline{t_i} \supset \psi) \in S$ or $(w, \sigma')T(\underline{t_i} \supset \theta) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. If $(w, \sigma')F(\underline{t_i} \supset \psi) \in S$ then, by the induction hypothesis applied to ψ, $t_i \nleq V(w, \psi)$. But clearly this contradicts $t_i = a \wedge V(w, \psi) \leq V(w, \psi)$. So we must have $(w, \sigma')F(\underline{a} \supset \psi) \notin S$
Hence, $(w, \sigma')T(\underline{t_i} \supset \theta) \in S$. So, by the induction hypothesis applied to θ, we have $t_i = a \wedge V(w, \psi) \leq V(w, \theta)$. Or equivalently, $T(\underline{a} \supset (\psi \supset \theta))$ is satisfied by $\mathfrak{M}_S$ at w.

<u>Case 1.2</u> β is of the form $F(\underline{a} \supset (\psi \supset \theta))$.
Suppose $(w, \sigma)F(\underline{a} \supset (\psi \supset \theta)) \in S$. Condition (2) implies $a \neq 0$. Hence, Condition (7) implies that there exists some $0 < t_i \leq a$ where $(w, \sigma')T(\underline{t_i} \supset \psi) \in S$ and $(w, \sigma')F(\underline{t_i} \supset \theta) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. So, by the induction hypothesis, $t_i \leq V(w, \psi)$ and $t_i \nleq V(w, \theta)$.
Since $t_i \leq a$ and $t_i \leq V(w, \psi)$, we have $t_i \leq a \wedge V(w, \psi)$. Note that if $a \wedge V(w, \psi) \leq V(w, \theta)$, then $t_i \leq V(w, \theta)$ (by the transitivity of $\leq$). Therefore, since $t_i \nleq V(w, \theta)$, we must have $a \wedge V(w, \psi) \nleq V(w, \theta)$. Or equivalently, $F(\underline{a} \supset (\psi \supset \theta))$ is satisfied by $\mathfrak{M}_S$ at w.

Case 1.3 β is of the form $T((\psi \supset \theta) \supset \underline{a})$.
Suppose $(w,\sigma)T((\psi \supset \theta) \supset \underline{a}) \in S$. Then, Condition (16) implies that, for all $u \in min(\{c \in H \mid c \nleq a\})$, $(w,\sigma')F(\underline{u} \supset (\psi \supset \theta)) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Now, by applying the same argument used above in Case 1.2, we can establish that $u \nleq V(w, \psi \supset \theta)$ for all $u \in min(\{c \in H \mid c \nleq a\})$. Thus, by Lemma 2.11 (2), we conclude that $V(w, \psi \supset \theta) \leq a$. Or equivalently, $T((\psi \supset \theta) \supset \underline{a})$ is satisfied by $\mathfrak{M}_S$ at w.

Case 1.4 β is of the form $F((\psi \supset \theta) \supset \underline{a})$.
Suppose $(w,\sigma)F((\psi \supset \theta) \supset \underline{a}) \in S$. By Condition (2), we must have $a \neq 1$. Then, Condition (14) implies $(w,\sigma')T(\underline{u} \supset (\psi \supset \theta)) \in S$ for some $u \in min(\{c \in H \mid c \nleq a\})$ and $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Now, by applying the same argument used above in Case 1.1, we can establish that $u \leq V(w, \psi \supset \theta)$. Thus, by the contrapositive of Lemma 2.11 (2), we conclude that $V(w, \psi \supset \theta) \nleq a$. Or equivalently, $F((\psi \supset \theta) \supset \underline{a})$ is satisfied by $\mathfrak{M}_S$ at w.

We now consider the modal cases.

- φ is of the form $\Box\psi$
  Let $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$, $a \in H$ and β a signed bounding implication that bounds φ by a. So β is of the form $T(\underline{a} \supset \Box\psi), T(\Box\psi \supset \underline{a}), F(\underline{a} \supset \Box\psi)$ or $F(\Box\psi \supset \underline{a})$. We consider each case.

Case 2.1 β is of the form $T(\underline{a} \supset \Box\psi)$.
Suppose $(w,\sigma)T(\underline{a} \supset \Box\psi) \in S$. Note that $T(\underline{a} \supset \Box\psi)$ is satisfied by $\mathfrak{M}_S$ at w iff for all $v \in W$,

$$a \leq R(w,v) \Rightarrow V(v,\psi). \tag{4.15.1}$$

Let $v \in W \subseteq \Sigma$. If $R(w,v) = 0$, then (4.15.1) clearly holds. So assume $R(w,v) \neq 0$. Then $R(w,v) = R_S(w,v)$ and $(w, v, \underline{R(w,v)}) \in cons(S)$. Hence, by Condition (9), we have $(v,\sigma')T(\underline{a \wedge R(w,v)} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Thus, by the induction hypothesis applied to ψ, we have $a \wedge R(w,v) \leq V(v,\psi)$. Or equivalently, (4.15.1) holds.

Case 2.2 β is of the form $F(\underline{a} \supset \Box\psi)$.
Suppose $(w,\sigma)F(\underline{a} \supset \Box\psi) \in S$. By Condition (11), there exists some $v \in \Sigma$ and $t_i \in H$ such that $a \wedge t_i \neq 0$, $(w, v, \underline{t_i}) \in cons(S)$ and $(v,\sigma')F(\underline{t_i \wedge a} \supset \psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. So $R(w,v) = R_S(w,v) = t_i$ and by the induction hypothesis applied to ψ, $F(\underline{R(w,v) \wedge a} \supset \psi)$ is satisfied by $\mathfrak{M}_S$ at v. Therefore, $a \nleq R(w,v) \Rightarrow V(v,\psi)$ and thus we must have

$$\begin{aligned} a &\nleq \bigwedge\{R(w,v) \Rightarrow V(v,\psi) \mid v \in W\} \\ &= V(w,\Box\psi). \end{aligned}$$

Or equivalently, $F(\underline{a} \supset \Box\psi)$ is satisfied by $\mathfrak{M}_S$ at w.

Case 2.3 β is of the form $T(\Box\psi \supset \underline{a})$.
Suppose $(w,\sigma)T(\Box\psi \supset \underline{a}) \in S$. Then, Condition (16) implies that for all $u \in min(\{c \in H \mid c \nleq a\})$, $(w,\sigma')F(\underline{u} \supset \Box\psi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Now, by applying the same argument used above in Case 2.2, we can establish that $u \nleq V(w,\Box\psi)$ for all such u. Thus, by Lemma 2.11 (2), we conclude that $V(w,\Box\psi) \leq a$. Or equivalently, $T(\Box\psi \supset \underline{a})$ is satisfied by $\mathfrak{M}_S$ at w.

Case 2.4 β is of the form $F(\Box\psi \supset \underline{a})$.
Suppose $(w,\sigma)F(\Box\psi \supset \underline{a}) \in S$. By Condition (2), we must have $a \neq 1$. Then, Condition (14) implies $(w,\sigma')T(\underline{u} \supset \Box\psi) \in S$ for some $u \in min(\{c \in H \mid c \nleq a\})$ and $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Now, by

applying the same argument used above in Case 2.1, we can establish that $u \leq V(w, \Box\psi)$. Thus, by the contrapositive of Lemma 2.11 (2), we conclude that $V(w, \Box\psi) \not\leq a$. Or equivalently, $F(\Box\psi \supset \underline{a})$ is satisfied by $\mathfrak{M}_S$ at w.

- φ is of the form $\Diamond\psi$
  Let $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$, $a \in H$ and β a signed bounding implication that bounds φ by a. So β is of the form $T(\underline{a} \supset \Diamond\psi), T(\Diamond\psi \supset \underline{a}), F(\underline{a} \supset \Diamond\psi)$ or $F(\Diamond\psi \supset \underline{a})$. We consider each case.

Case 3.1 β is of the form $T(\Diamond\psi \supset \underline{a})$.
Suppose $(w, \sigma)T(\Diamond\psi \supset \underline{a}) \in S$. Note that $T(\Diamond\psi \supset \underline{a})$ is satisfied by $\mathfrak{M}_S$ at w iff for all $v \in W$,

$$R(w, v) \wedge V(v, \psi) \leq a. \tag{4.15.2}$$

Let $v \in W \subseteq \Sigma$. If $R(w, v) = 0$, then (4.15.2) clearly holds. So assume $R(w, v) \neq 0$. Then, $R(w, v) = R_S(w, v)$ and $(w, v, \underline{R(w, v)}) \in cons(S)$. Hence, by Condition (10), we have $(v, \sigma')T(\psi \supset \underline{R(w, v) \Rightarrow a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Thus, by the induction hypothesis applied to ψ, we have $V(v, \psi) \leq R(w, v) \Rightarrow a$. Or equivalently, (4.15.2) holds.

Case 3.2 β is of the form $F(\Diamond\psi \supset \underline{a})$.
Suppose $(w, \sigma)F(\Diamond\psi \supset \underline{a}) \in S$. By Condition (12), there exists some $v \in \Sigma$ and $t_i \in H$ such that $t_i \Rightarrow a \neq 1$, $(w, v, \underline{t_i}) \in cons(S)$ and $(v, \sigma')F(\psi \supset \underline{t_i \Rightarrow a}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. So $R(w, v) = R_S(w, v) = t_i$ and by the induction hypothesis, $F(\psi \supset \underline{R(w, v) \Rightarrow a})$ is satisfied by $\mathfrak{M}_S$ at v. Therefore, $R(w, v) \wedge V(v, \psi) \not\leq a$ and thus we must have

$$\begin{aligned} a &\not\geq \bigvee\{R(w, v) \wedge V(v, \psi) \mid v \in W\} \\ &= V(w, \Diamond\psi). \end{aligned}$$

Or equivalently, $F(\underline{a} \supset \Diamond\psi)$ is satisfied by $\mathfrak{M}_S$ at w.

Case 3.3 β is of the form $T(\underline{a} \supset \Diamond\psi)$.
Suppose $(w, \sigma)T(\underline{a} \supset \Diamond\psi)$. Then, Condition (15) implies that for all $t \in max(\{c \in H \mid a \not\leq c\})$, $(w, \sigma')F(\Diamond\psi \supset \underline{t}) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. Now, by applying the same argument used above in Case 3.2, we can establish that $V(w, \Diamond\psi) \not\leq t$ for all such t. Thus, by Lemma 2.11 (1), we conclude that $a \leq V(w, \Diamond\psi)$. Or equivalently, $T(\underline{a} \supset \Diamond\psi)$ is satisfied by $\mathfrak{M}_S$ at w.

Case 3.4 β is of the form $F(\underline{a} \supset \Diamond\psi)$.
Suppose $(w, \sigma)F(\underline{a} \supset \Diamond\psi) \in S$. By Condition (2), we must have $a \neq 0$. Then, Condition (13) implies $(w, \sigma)T(\Diamond\psi \supset \underline{t}) \in S$ for some $t \in max(\{c \in H \mid a \not\leq c\})$. Now, by applying the same argument used above in Case 3.1, we can establish that $V(w, \Box\psi) \leq t$. Thus, by the contrapositive of Lemma 2.11 (1), we conclude that $a \not\leq V(w, \Diamond\psi)$. Or equivalently, $F(\underline{a} \supset \Diamond\psi)$ is satisfied by $\mathfrak{M}_S$ at w.

This concludes the proof that for every formula φ, $P(\varphi)$ holds. To show that this implies S is staisfiable, consider the identity map $I : W \to W$. Clearly, by Condition 1 of Definition 4.13 and the construction of R, we have that I is an interpretation of S in $\mathfrak{M}_S$.
Suppose $(w, \sigma)\beta \in S$ for some $w \in \Sigma$ and $\sigma \subset \Sigma^2 \times \underline{H}$, where β is a signed bounding implication. For some $a \in H$, β must bound some formula φ by a. But since $P(\varphi)$ holds, we can conclude that β is satisfied by $\mathfrak{M}_S$ at $w = I(w)$.
Thus, S is satisfied under I. $\square$

4.2.1 Decision Procedure

We now finally present the decision procedure. Essentially, it amounts to constructing a tableau by systematically applying rules until either we have a closed tableau or a tableau in which a downward saturated branch exists. We use a labelled tree as the data structure representing the tableau (Refer to Definition 4.4 and 4.6).

---

procedure isValid(φ) returns **true** or **false**

Require: formula φ

$\alpha \coloneqq F(\underline{1} \supset \varphi)$
$(\mathscr{T}, U) \coloneqq$ constructTableau(α)
if $(\mathscr{T}, U)$ is closed **then**
    return true
else
    return false
end if

---

constructTableau(α) returns a labelled tree $(\mathscr{T}, U)$

Require: Signed bounding implication α

Initialize a labeled tree $(\mathscr{T}, U)$ with root node r and $U(r) \coloneqq (w_0, \varnothing)\alpha$ ▷ Pick any $w_0 \in \Sigma$
Mark r as being unfinished ▷ From now on we will assume that any newly created node is marked as unfinished by default
$i \coloneqq 0$
$(\mathscr{T}_i, U_i) \coloneqq (\mathscr{T}, U)$
while there are unfinished nodes **and** $(\mathscr{T}, U)$ is not closed **do**
    Pick some unfinished node n and mark it as finished.
    Assume $U(n) = (w, \sigma)\beta$
    for each open branch $\mathscr{B}$ of $\mathscr{T}_i$ containing n **do**
        We now proceed to extend or fork $\mathscr{B}$ depending on the form of $U(n)$.
        In what follows, assume we only add a node labelled with $(u, \sigma')\beta'$ if $(u, \sigma'')\beta' \notin U(\mathscr{B})$ for all σ''
        Assume l is the leaf of $\mathscr{B}$.
        Let $\sigma' = cons(U(\mathscr{B}))$
        if $U(\mathscr{B})$ is an instantiation of the numerator of the rule $p\bot_1, p\bot_2, p\bot_3, p\bot_4$ or $p\bot_5$ **then**
            Extend $\mathscr{B}$ with a node labelled $(w, \varnothing)\bot$. ▷ By extending a path we mean adding a node to the end of it
            Continue to the next iteration

---

else if β is $F(\underline{a} \supset \varphi)$ where $a \in H$ and φ is of the form p (a propositional variable) **or** $\psi \vee \theta$ **or** $\Diamond\psi$ **then**
  for each $t \in max(\{c \in H \mid a \not\leq c\})$ **do**
    Create a node n', with $U(n') = (w, \sigma')T(\varphi \supset \underline{t})$
    Add n' as a child of l
  end for
else if β is $F(\varphi \supset \underline{a})$ where $a \in H$, $a \neq 1$ and φ is of the form p (a propositional variable) **or** $\psi \wedge \theta$ **or** $\psi \supset \theta$ **or** $\Box\psi$ **then**
  for each $u \in min(\{c \in H \mid c \not\leq a\})$ **do**
    Create a node n', with $U(n') = (w, \sigma')T(\underline{u} \supset \varphi)$
    Add n' as a child of l
  end for
else if β is $T(\underline{a} \supset \varphi)$ where $a \in H$ and φ is of the form $\psi \vee \theta$ **or** $\Diamond\psi$ **then**
  for each $t \in max(\{c \in H \mid a \not\leq c\})$ **do**
    Create a new node n' with $U(n') = (w, \sigma')F(\varphi \supset \underline{t})$
    Extend $\mathscr{B}$ with n'
  end for
else if β is $T(\varphi \supset \underline{a})$ where $a \in H$ and φ is of the form $\psi \wedge \theta$ **or** $\psi \supset \theta$ **or** $\Box\psi$ **then**
  for each $u \in min(\{c \in H \mid c \not\leq a\})$ **do**
    Create a new node n' with $U(n') = (w, \sigma')F(\underline{u} \supset \varphi)$
    Extend $\mathscr{B}$ with n'
  end for
else if β is of the form $T(\underline{a} \supset (\varphi \wedge \psi))$ for some truth value $a \neq 0$ **then**
  Create nodes n' and n'', with $U(n') = (w, \sigma')T(\underline{a} \supset \varphi)$ and $U(n'') = (w, \sigma')T(\underline{a} \supset \psi)$
  Extend $\mathscr{B}$ with n' and n''
else if β is of the form $F(\underline{a} \supset (\varphi \wedge \psi))$ for some truth value $a \neq 0$ **then**
  Create nodes n' and n'', with $U(n') = (w, \sigma')F(\underline{a} \supset \varphi)$ and $U(n'') = (w, \sigma')F(\underline{a} \supset \psi)$
  Add n' and n'' as children of l

else if β is of the form $T((\varphi \vee \psi) \supset \underline{a})$ for some truth value $a \neq 1$ **then**
Create nodes n' and n'', with
$U(n') = (w, \sigma')T(\varphi \supset \underline{a})$ and $U(n'') = (w, \sigma')T(\psi \supset \underline{a})$
Extend $\mathscr{B}$ with n' and n''
else if β is of the form $F((\varphi \vee \psi) \supset \underline{a})$ for some truth value $a \neq 1$ **then**
Create nodes n' and n'', with
$U(n') = (w, \sigma')F(\varphi \supset \underline{a})$ and $U(n'') = (w, \sigma')F(\psi \supset \underline{a})$
Add n' and n'' as children of l
else if β is of the form $F(\underline{a} \supset (\varphi \supset \psi))$ **then**
for each $t \in \{c \in H \mid c \leq a \text{ and } c \neq 0\}$ **do**
Create nodes n' and n'', with
$U(n') = (w, \sigma')T(\underline{t} \supset \varphi)$ and $U(n'') = (w, \sigma')F(\underline{t} \supset \psi)$
Add n' as a child of l and n'' as a child of n'
end for
else if β is of the form $T(\underline{a} \supset (\varphi \supset \psi))$ **then**
for each $t \in \{c \in H \mid c \leq a \text{ and } c \neq 0\}$ **do**
Create nodes n' and n'', with
$U(n') = (w, \sigma')F(\underline{t} \supset \varphi)$ and $U(n'') = (w, \sigma')T(\underline{t} \supset \psi)$
if this is the first iteration of this for loop **then**
Add n' and n'' as children of l
else
for each of the nodes m added in the previous iteration of this for loop **do**
Add copies of n' and n'' as children of m
end for
end if
end for
else if β is of the form $T(\underline{a} \supset \Box\varphi)$ **then**
for each $v \in \Sigma$ and $t \in H$ such that $(w, v, \underline{t}) \in \sigma'$ **do**
Create a new node n' with $U(n') = (v, \sigma')T(\underline{a \wedge t} \supset \varphi)$
Extend $\mathscr{B}$ with n'
end for
else if β is of the form $T(\Diamond\varphi \supset \underline{a})$ **then**
for each $v \in \Sigma$ and $t \in H$ such that $(w, v, \underline{t}) \in \sigma'$ **do**
Create a new node n' with $U(n') = (v, \sigma')T(\varphi \supset \underline{t \Rightarrow a})$
Extend $\mathscr{B}$ with n'
end for

else if β is of the form $F(\underline{a} \supset \Box\varphi)$ **then**
Pick some $v \in \Sigma$ that does not already occur in $worlds(U(\mathscr{B}))$.
for each $t \in H$ such that $a \wedge t \neq 0$ **do**
Create a new node n' with $U(n') = (v, \sigma' \cup \{(w, v, \underline{t})\})F(\underline{a \wedge t} \supset \varphi)$
Add n' as a child of l
end for
REACTIVATE(n)
else if β is of the form $F(\Diamond\varphi \supset \underline{a})$ **then**
Pick some $v \in \Sigma$ that does not already occur in $worlds(U(\mathscr{B}))$.
for each $t \in H$ such that $t \Rightarrow a \neq 1$ **do**
Create a new node n' with $U(n') = (v, \sigma' \cup \{(w, v, \underline{t})\})F(\varphi \supset \underline{t \Rightarrow a})$
Add n' as a child of l
end for
REACTIVATE(n)
end if
end for
Increment i by 1
$(\mathscr{T}_i, U_i) := (\mathscr{T}, U)$
end while
return $(\mathscr{T}, U)$

REACTIVATE(n)

Require: Node n
Assume $U(n) = (w, \sigma)\beta$
for each open branch $\mathscr{B}'$ of $\mathscr{T}$ containing n **do**
Let $\sigma' = cons(U(\mathscr{B}'))$
for each finished node m in $\mathscr{B}'$ **do**
if $sf(U(m))$ is of the form $T(\underline{a} \supset \Box\varphi)$ **then**
for each $v \in \Sigma$ and $t \in H$ such that $(w, v, \underline{t}) \in \sigma'$ **do**
Create a new node n' with $U(n') = (v, \sigma')T(\underline{a \wedge t} \supset \varphi)$
Extend $\mathscr{B}'$ with n'
end for
else if $sf(U(m))$ is of the form $T(\Diamond\varphi \supset \underline{a})$ **then**
for each $v \in \Sigma$ and $t \in H$ such that $(w, v, \underline{t}) \in \sigma'$ **do**
Create a new node n' with $U(n') = (v, \sigma')T(\varphi \supset \underline{t \Rightarrow a})$
Extend $\mathscr{B}'$ with n'
end for
end if
end for
end for

Remark 4.16. $(\mathscr{T}_i, U_i)$ denotes the labeled tree $(\mathscr{T}, U)$ after the i^{th} iteration of the while loop. In other words, $(\mathscr{T}_i, U_i)$ is a snapshot of the continuously growing labeled tree $(\mathscr{T}, U)$, and there may be moments during the course of execution of the for loop on line 8 where they are not the same thing (as for instance, will often be the case whenever we reach line 2 in REACTIVATE).

Example 4.17. The pretty daunting length of CONSTRUCTTABLEAU should not be cause for alarm. After a bit of consideration, we see that this length is largely due to the numerous cases we have to consider, where each case repeats a theme: we are applying rules to a branch of the labeled tree with the aim of making a specific condition of Definition 4.13 hold for the set of labels in that branch.
Consider $\varphi \equiv \Box p \supset \Box\Diamond p$. ISVALID($\varphi$) returns **false**. Let us see why by going through the steps of the procedure. Line 2 of ISVALID(φ) invokes CONSTRUCTTABLEAU$F(\underline{1} \supset \varphi)$, which returns the labeled tree shown in Figure 4.3.
We trace through the iterations of the while loop in CONSTRUCTTABLEAU$F(\underline{1} \supset \varphi)$:

0. $(\mathscr{T}, U)$ is used throughout the procedure to denote the current state of a labeled tree that will grow as we progress. Line 1 of CONSTRUCTTABLEAU initializes $(\mathscr{T}, U)$ to consist of only node 1, which is labeled with $(\mathtt{A}, \varnothing)F(\underline{1} \supset \varphi)$ (we pick $w_0 = \mathtt{A} \in \Sigma$), and node 1 is marked as unfinished in line 2. This concludes the 0^{th} iteration of the while loop and $(\mathscr{T}_0, U_0)$ is set to the current state of $(\mathscr{T}, U)$. Clearly $T_{(\mathscr{T}_0, U_0)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A}, \varnothing)F(\underline{1} \supset \varphi)\}$.

1. The branch $\mathscr{B}_0^1$ containing only node 1 is a branch of $\mathscr{T}_0$ (Note, in this example we shall use $\mathscr{B}_i^j$ to denote the branch of tree $\mathscr{T}_i$ with leaf node j). Node 1 is unfinished and clearly $\mathscr{B}_0^1$ is open, so we enter the 1^{st} iteration of the while loop. In line 6 we pick node 1 and then line 7 amounts to setting $w = \mathtt{A}, \sigma = \varnothing, \beta = F(\underline{1} \supset \varphi)$, according to the label of node 1. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_0^1$, which is the only open branch of $\mathscr{T}_0$ containing node 1. On line 12 we set $\sigma' = cons(\mathscr{B}) = \varnothing$.
   The **if** condition on line 48 is met. So, the steps in lines 49 to 52 are performed. This amounts to adding nodes 2, 3, 4 and 5, which reflects an application of $pF\supset$ to $\mathscr{T}$. We now return to line 8, the beginning of the for loop over open branches of $\mathscr{T}_0$ [10] containing node 1. However, there are no other open branches of $\mathscr{T}_0$ containing node 1 left to check, so we exit the for loop. This ends the 1^{st} iteration of the while loop, and line 91 sets $(\mathscr{T}_1, U_1)$ to the current state of $(\mathscr{T}, U)$.

2. We return to the start of the while loop at line 5. $(\mathscr{T}_1, U_1)$ consists of the unfinished nodes 2, 3, 4 and 5, and the open branches $\mathscr{B}_1^3$ and $\mathscr{B}_1^5$. So we enter the 2^{nd} iteration of the while loop. Suppose we pick node 2 in line 6. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_1^3$, which is the only open branch of $\mathscr{T}_1$ containing node 2. Line 12 sets $\sigma' = cons(\mathscr{B}) = \varnothing$. The **if** condition on line 64 is met. So, the steps in lines 65 to 68 are performed. However, since σ' is empty, no new nodes are added. Looked at another way, we have performed an identity application of $p\mathbf{K}T\Box$. We now return to line 8 and exit the for loop. This ends the 2^{nd} iteration of the while loop, and line 91 sets $(\mathscr{T}_2, U_2)$ to the current state of $(\mathscr{T}, U)$.

3. We return to the start of the while loop at line 5. $(\mathscr{T}_2, U_2)$ consists of the unfinished nodes 3, 4 and 5, and the open branches $\mathscr{B}_2^3$ and $\mathscr{B}_2^5$. So we enter the 3^{rd} iteration of the while loop. Suppose we pick node 3 in line 6. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_2^3$, which is the only open branch of $\mathscr{T}_2$ containing node 3. Line 12 sets $\sigma' = cons(\mathscr{B}) = \varnothing$. The **if** condition on line 74 is met. So, the steps in lines 75 to 80 are performed. Lines 75 to 79 amount to an application of

[10] Note that we are concerned with branches in $\mathscr{T}_i$, not those in $\mathscr{T}$, which may be different at some point of the i^{th} iteration.

$p\mathbf{K}F\Box$, which adds nodes 6 and 7 to $\mathscr{T}$. In line 80, REACTIVATE is called on node 3. In essence, REACTIVATE ensures that, after a new constraint is added to a branch, any previous applications of $p\mathbf{K}T\Box$ and $p\mathbf{K}T\Diamond$ that were applied to the branch are 'reactivated' so as to ensure that Conditions (9) and (10) of downward saturation are maintained. Let us see what this means in the current context. After applying $p\mathbf{K}F\Box$, we have that the branch of $\mathscr{T}$ with leaf node 6 and the branch of $\mathscr{T}$ with leaf node 7 are open branches containing node 3. Now node 2 was marked finished in iteration 2 of the while loop. So, executing REACTIVATE on node 3 amounts to applying $p\mathbf{K}T\Box$ to each of these branches, around the principle element being the label of node 2. This leads to nodes 8 and 9 being added to $\mathscr{T}$. REACTIVATE is then exited. The 3^{rd} iteration of the while loop is ended, and line 91 sets $(\mathscr{T}_3, U_3)$ to the current state of $(\mathscr{T}, U)$.

4. We return to the start of the while loop at line 5. $(\mathscr{T}_3, U_3)$ consists of the unfinished nodes 4, 5, 6, 7, 8, and 9, and the open branches $\mathscr{B}_3^8$, $\mathscr{B}_3^9$ and $\mathscr{B}_3^5$. So we enter the 4^{th} iteration of the while loop. Suppose we pick node 6 in line 6. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_3^8$, which is the only open branch of $\mathscr{T}_3$ containing node 6. Line 12 sets $\sigma' = cons(\mathscr{B}) = \{(\mathtt{A}, \mathtt{B}, \underline{1})\}$. The **if** condition on line 16 is met. So, the steps in lines 17 to 20 are performed, in which node 10 is added to $\mathscr{T}$. This amounts to an application of $pF\geq$. This ends the 4^{th} iteration of the while loop, and line 91 sets $(\mathscr{T}_4, U_4)$ to the current state of $(\mathscr{T}, U)$.

5. We return to the start of the while loop at line 5. $(\mathscr{T}_4, U_4)$ consists of the unfinished nodes 4, 5, 7, 8, 9 and 10, and the open branches $\mathscr{B}_4^{10}$, $\mathscr{B}_4^9$ and $\mathscr{B}_4^5$. So we enter the 5^{th} iteration of the while loop. Suppose we pick node 8 in line 6. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_4^{10}$, which is the only open branch of $\mathscr{T}_4$ containing node 8. Line 12 sets $\sigma' = cons(\mathscr{B}) = \{(\mathtt{A}, \mathtt{B}, \underline{1})\}$. None of the **if** conditions are met, so no nodes are added to $\mathscr{T}$. This ends the 5^{th} iteration of the while loop, and line 91 sets $(\mathscr{T}_5, U_5)$ to the current state of $(\mathscr{T}, U)$.

6. We return to the start of the while loop at line 5. $(\mathscr{T}_5, U_5)$ consists of the unfinished nodes 4, 5, 7, 9 and 10, and the open branches $\mathscr{B}_5^{10}$, $\mathscr{B}_5^9$ and $\mathscr{B}_5^5$. So we enter the 6^{th} iteration of the while loop. Suppose we pick node 10 in line 6. We then enter the for loop on line 8, and set $\mathscr{B} = \mathscr{B}_5^{10}$, which is the only open branch of $\mathscr{T}_5$ containing node 10. Line 12 sets $\sigma' = cons(\mathscr{B}) = \{(\mathtt{A}, \mathtt{B}, \underline{1})\}$. The **if** condition on line 69 is met. However, for $w = \mathtt{B}$, there are no $(w, v, \underline{t}) \in \sigma'$. So, no new nodes are added to $\mathscr{T}$ (that is, we perform an identity application of $p\mathbf{K}T\Diamond$). This ends the 6^{th} iteration of the while loop, and line 91 sets $(\mathscr{T}_6, U_6)$ to the current state of $(\mathscr{T}, U)$.

We carry on in this manner, picking unfinished nodes, until either no unfinished nodes are left or $(\mathscr{T}, U)$ is closed. Consider the branch $\mathscr{B} = \mathscr{B}_6^{10}$. Notice that all the nodes in this branch have been finished after iteration 6, and so no further iterations of the while will change this branch. Hence, this branch will be present in the final labeled tree returned by CONSTRUCTTABLEAU$F(\underline{1} \supset \varphi)$, and this is what leads ISVALID(φ) to return **false**.

And in fact, $U(\mathscr{B})$ is downward saturated (A fact regarding open labeled trees constructed by our procedure that will be proven in general for Proposition 4.23). As in the proof of Lemma 4.15, $U(\mathscr{B})$ induces the following $\mathcal{H}^3$-model:

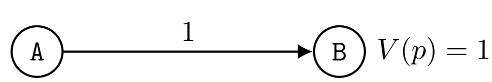


Figure 4.2

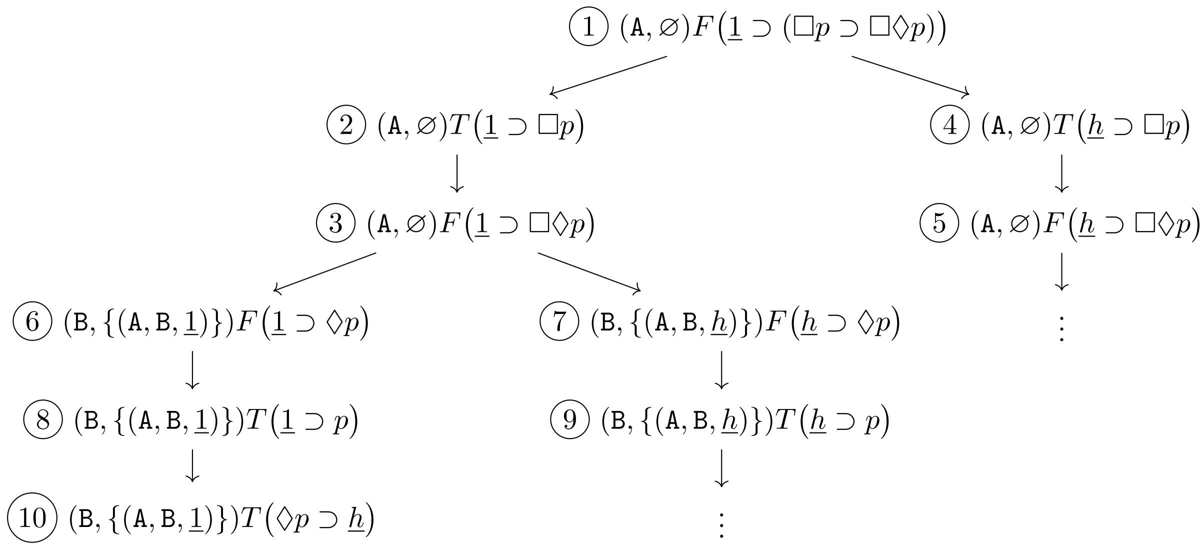


Figure 4.3

As the reader can confirm, the value of φ at $\mathtt{A}$ is 0. And so, this model is a countermodel for φ.

Also, observe that after each iteration i of the while loop, $(\mathscr{T}_i, U_i)$ has resulted from a finite sequence of $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-rule applications. As such, after termination, $T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(\mathtt{A}, \varnothing)F(\underline{1} \supset \varphi)\}$. As we shall see, this observation is a special case of Lemma 4.22.

Remark 4.18. Note that the step in line 6 of constructTableau is nondeterministic in the sense that there may be multiple unfinished nodes to pick from. Any method of picking such a node will yield a terminating and correct procedure. However, they are not all equally efficient, since the unfinished node we pick at a given stage can dramatically influence the size of the constructed tableau. For instance, it could be favorable to first pick nodes that lead to the application of rules with the fewest denominators (and hence least forking of the tree).

Remark 4.19. No branch of $\mathscr{T}$ is ever shrunk during the execution of constructTableau.
Further, let $\mathscr{B}$ be a branch of the constructed tree. For all $i \in \mathbb{N}$, if the node n was added to $\mathscr{B}$ during the i^{th} iteration, for every node n' added to $\mathscr{B}$ in iteration $j \leq i$,

$$con(U(n')) \subseteq con(U(n)).$$

Proposition 4.20. *For all formulas φ, isValid(φ) terminates.*

Proof. Consider an arbitrary formula φ and assume isValid(φ) does not terminate. We derive a contradiction. isValid(φ) does not terminate only if the while loop in constructTableau$(F(\underline{1} \supset \varphi))$ goes on forever. And this can only be the case if we are constructing an infinite tree $\mathscr{T}$ (since each iteration of the while loop marks an unfinished node as finished, and only new nodes are set as unfinished).
Each tableau rule has only a finite number of denominators, and so it is not hard to see that $\mathscr{T}$ is finitely generated. Thus, by König's Lemma 2.76, $\mathscr{T}$ must have an infinite branch $\mathscr{B}$.

The procedure only adds a node to a branch if its label does not already occur in that branch (see line 10). Hence, $U(\mathscr{B})$ must be infinite. Further, it should be noted that for every node n' added to $\mathscr{B}$, we have that $sf(U(n'))$ is a signed bounded subformula (recall Definition 2.32) of φ and so,

$$sf(x) \text{ is a signed bounded subformula of } \varphi \text{ for all } x \in U(\mathscr{B}) \tag{4.20.1}$$

For each $k \in \mathbb{N}$, define

$$A_k \coloneqq \{x \in U(\mathscr{B}) \mid con(x) \text{ has at most } k \text{ elements}\},$$

$$B_k \coloneqq \{x \in U(\mathscr{B}) \mid con(x) \text{ has exactly } k \text{ elements}\}.$$

Firstly, we argue by induction that for every $k \in \mathbb{N}$,

$$|worlds(A_k)| \leq k + 1 \tag{4.20.2}$$

The base case requires us to show that $|worlds(A_0)| \leq 1$. For the root node r, we have that $U(r) = (w_0, \varnothing)F(\underline{1} \supset \varphi) \in A_0$. Consider $C \coloneqq \{x \in U(\mathscr{B}) \mid p(x) = (w_0, \varnothing)\}$.
Clearly $C \subseteq A_0$. To see that $A_0 \subseteq C$, suppose $x \notin C$ and that, without loss of generality, $x \in U(\mathscr{B})$ and $world(x) = v$ for some $v \neq w_0$. Then, it must be the case that a node n' was added to $\mathscr{B}$ through an application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ (by line 75 or 82) during the j^{th} iteration, where $p(U(n')) = (v, \sigma' \cup \{(w, v, \underline{t})\})$ for some $\sigma \subseteq \Sigma^2 \times \underline{H}$, $w \in \Sigma$ and $t \in H$. And for some $i \geq j$, a node n with $U(n) = x$ was added to $\mathscr{B}$ during the i^{th} iterartion. Thus, by Remark 4.19, $\sigma' \cup \{(w, v, \underline{t})\} \subseteq con(x)$ and therefore $x \notin A_0$. Hence, $C = A_0$, and $|worlds(A_0)| = |worlds(C)| = |\{w_0\}| = 1$.

Now, let $k \geq 1$ and, as our induction hypothesis, assume (4.20.2) holds. Let $v \in \Sigma$ and suppose $v \in worlds(A_{k+1})$ but $v \notin worlds(A_k)$. Then v must have been introduced via an application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$. But since such applications also introduce new constraints to the branch, and by Remark 4.19, we must have that v is the unique world label such that $v \in worlds(A_{k+1})$ and $v \notin worlds(A_k)$.
Thus, $worlds(A_{k+1}) \subseteq worlds(A_k) \cup \{v\}$, and hence $|worlds(A_{k+1})| \leq |worlds(A_k)| + 1$. So, by the induction hypothesis, $|worlds(A_{k+1})| \leq k + 2$, as required.

Consider an arbitrary $k \in \mathbb{N}$. We proceed to show that B_k is finite. By the previous induction, $worlds(A_k)$ is finite. Hence, $worlds(B_k)$ (which is a subset of $worlds(A_k)$) is finite.
Let $x, x' \in B_k'$. So,

$$|con(x)| = |con(x')| \tag{4.20.3}$$

where $x = U(n)$ and $x' = U(n')$ for some nodes n, n' in $\mathscr{B}$. Without loss of generality, suppose n was added to $\mathscr{B}$ after n'. Then, by remark 4.19, $con(x') \subseteq con(x)$. So by (4.20.3), we must have $con(x) = con(x')$.
Thus, $con(x)$ is the same for every $x \in B_k$; call it σ_k. We have $x \in B_k$ iff x is of the form $(w, \sigma_k)\beta$ where $w \in worlds(B_k)$ and β is a signed bounded subformula of φ (by 4.20.1). There are only finitely many such x. Thus, B_k must be finite.

Recall Remark 4.18. For the sake of simplifying this proof, let us assume that we pick an unfinished node with a label that has the maximum $Mdegree$ (recall Definition 2.29) among unfinished nodes. Under this assumption, it is not too hard to see that as k increases,

$$\sum_{x \in B_k} Mdegree(x)$$

decreases. Thus, there must exist some k for which all elements of B_k have $Mdegree$ 0. But this means that $B_{k'} = \varnothing$ for all $k' > k$. Therefore,

$$U(\mathscr{B}) = B_0 \cup \ldots \cup B_k,$$

where $B_0, \ldots, B_k$ are each finite. And so $U(\mathscr{B})$ must be finite, which is contrary to what we established earlier. $\square$

The following remark introduces concepts and terms that will be useful in the arguments to come.

Remark 4.21. Let $i, j \in \mathbb{N}$ and suppose $i \leq j$.
We have $U_i \subseteq U_j$ and so for all nodes n in $\mathscr{T}_i$, $U_i(n) = U_j(n)$. As such, we will usually just write $U(n)$, where U is the final labeling.
The next useful property follows from the fact that branches are only extended and/or split from the leaf node. For all branches $\mathscr{B}_j$ of $\mathscr{T}_j$, there exists a unique branch $\mathscr{B}_i$ of $\mathscr{T}_i$ such that $\mathscr{B}_i$ is a subpath of $\mathscr{B}_j$ starting at the root. And, $U(\mathscr{B}_i) \subseteq U(\mathscr{B}_j)$.
Let us denote by $\mathscr{P}_i$ the (possibly empty) path with which the end of $\mathscr{B}_{i-1}$ is extended in iteration i to get $\mathscr{B}_i$.

Lemma 4.22. *Let α be a signed bounding implication. For the labeled tree $(\mathscr{T}, U)$ returned by* CONSTRUCTTABLEAU(α)*, $T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\boldsymbol{K}^{\mathcal{H}}$-tableau for $\{(w_0, \varnothing)\alpha\}$.*

Proof. We will prove that this is a loop invariant for the while loop performed by CONSTRUCTTABLEAU(α). We proceed by induction to show that for all $i \in \mathbb{N}$,

$$T_{(\mathscr{T}_i,U_i)} \text{ is a } p\mathcal{C}\mathbf{K}^{\mathcal{H}}\text{-tableau for } \{(w_0, \varnothing)\alpha\} \tag{4.22.1}$$

For the base case, suppose 0 iterations have been performed. $\mathscr{T}_0$ only consists of the root node r, and $U(r) = (w_0, \varnothing)\alpha$. So, $\mathscr{T}_0$ contains the single branch $\mathscr{B}$ where $\mathscr{B}$ contains the single node r. But by the base case of Definition 2.57, $T_{(\mathscr{T}_0,U_0)} = \{U(\mathscr{B})\} = \{\{U(r)\}\} = \{\{(w_0, \varnothing)\alpha\}\}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0, \varnothing)\alpha\}$.

Now, the inductive case requires us to show that (4.22.1) is maintained by each iteration of the while loop. As our induction hypothesis, let $i \in \mathbb{N}$ and suppose (4.22.1) holds. We now need to argue that it holds for $i+1$. To do so thoroughly would require considering each computation path that could possibly be taken during this iteration, each corresponding to the different **if** conditions. The arguments for each case are quite similar. Essentially we show that $(\mathscr{T}_{i+1}, U_{i+1})$ results from successive applications of $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ rules to $(\mathscr{T}_i, U_i)$. We shall look at a subset of these cases in order to illustrate the general idea. Let n be the node that has been picked in iteration $i+1$, where $U(n) = (w, \sigma)\beta$. Now, if we were to be pedantic, we would have to perform an inner induction on j, where j indexes the open branch $\mathscr{B}_j$ of $\mathscr{T}_i$ containing n. More specifically, we would have to show that $(\mathscr{T}_i^j, U_i^j)$ (that is, $(\mathscr{T}, U)$ immediately after the j^{th} iterartion of the for loop in line 8) results from a finite sequence of applications of $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ rules. For $j = 0$, this follows immediately from the induction hypothesis on i. To keep things simple, let us just show the case $j = 1$. The general inductive case uses essentially the same argument. So consider the first open branch $\mathscr{B}_1 = \mathscr{B}$ of $\mathscr{T}_i$ containing n we pick in line 8.

- Suppose the condition in line 39 is met. That is, β is of the form $F(\underline{a} \supset (\varphi \wedge \psi))$ for some truth value $a \neq 0$.
  So $U(\mathscr{B}) \in T_{(\mathscr{T}_i,U_i)}$ is an instantiation of the numerator of rule $pF\wedge$. The steps in lines 40 and 41 are

performed, which extend $(\mathscr{T}_i^0, U_i^0) = (\mathscr{T}_i, U_i)$ to give the labeled tree $(\mathscr{T}_i^1, U_i^1)$. More precisely, these steps ensure that $(\mathscr{T}_i^1, U_i^1)$ is the same as $(\mathscr{T}_i, U_i)$, except that $\mathscr{B}$ is split into (and hence replaced by) two new branches $\mathscr{B}_{n'}$ and $\mathscr{B}_{n''}$ with leaf nodes n' and n'' respectively. And, the labels of n' and n'' are such that $U(\mathscr{B}_{n'})$ and $U(\mathscr{B}_{n''})$ are instantiations of the denominators of $pF\wedge$ corresponding to $U(\mathscr{B})$. By the induction hypothesis, $T_{(\mathscr{T}_i,U_i)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0,\varnothing)\alpha\}$. So, in other words, lines 40 and 41 amount to applying rule $pF\wedge$ to $\mathscr{B}$. Thus, $T_{(\mathscr{T}_i^1,U_i^1)}$ is derivable from $T_{(\mathscr{T}_i,U_i)}$ via an application of $pF\wedge$, hence $T_{(\mathscr{T}_i^1,U_i^1)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0,\varnothing)\alpha\}$.

- Suppose the condition in line 42 is met. That is, β is of the form $T((\varphi \vee \psi) \supset \underline{a})$ for some truth value $a \neq 1$.
  So $U(\mathscr{B}) \in T_{(\mathscr{T}_i,U_i)}$ is an instantiation of the numerator of rule $pT\vee$, and the steps in lines 43 and 44 amounts to an application of $pT\vee$ to $\mathscr{B}$.

- Suppose the condition in line 64 is met. That is, β is of the form $T(\underline{a} \supset \Box\varphi)$.
  So $U(\mathscr{B}) \in T_{(\mathscr{T}_i,U_i)}$ is an instantiation of the numerator of rule $p\mathbf{K}T\Box$. Now, lines 65 to 68 correspond to successive, cumulative applications of $p\mathbf{K}T\Box$, for each of the $(v,t) \in \Sigma \times H$ that satisfy the side condition of this rule.

- Suppose the condition in line 74 is met. That is, β is of the form $F(\underline{a} \supset \Box\varphi)$.
  So $U(\mathscr{B}) \in T_{(\mathscr{T}_i,U_i)}$ is an instantiation of the numerator of rule $p\mathbf{K}F\Box$. Now, lines 75 to 79 correspond to an application of $p\mathbf{K}F\Box$, and the call to REACTIVATE(n) in line 80 amounts to zero or more successive, cumulative applications of $p\mathbf{K}T\Box$ and/or $p\mathbf{K}T\Diamond$.

In general, after iterating over each open branch containing n, we have iteratively and cumulatively applied $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau rules, starting with $(\mathscr{T}_i, U_i)$ and ending with $(\mathscr{T}_{i+1}, U_{i+1})$. Further, since the procedure terminates, we can conclude that only a finite number of rules are applied. This concludes the induction argument.

Now, since the while loop terminates, the labeled tree returned by CONSTRUCTTABLEAU(α) is $(\mathscr{T}_k, U_k)$ for some $k \in \mathbb{N}$. And the required result follows from 4.22.1. ☐

Proposition 4.23. *For all formulas φ,* ISVALID*(φ) returns* ***true*** *iff φ is valid.*

Proof. For the forward implication, suppose ISVALID(φ) returns **true**. The step in line 1 performed by ISVALID(φ) sets $\alpha = F(\underline{1} \supset \varphi)$ and, by Lemma 4.22, CONSTRUCTTABLEAU(α) in line 2 returns $(\mathscr{T}, U)$ where $T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0,\varnothing)\alpha\} = \{(w_0,\varnothing)F(\underline{1} \supset \varphi)\}$. Since ISVALID($\varphi$) returns **true**, we must have that the condition in line 3 is met. That is, $(\mathscr{T}, U)$ is closed, and so $T_{(\mathscr{T},U)}$ is a closed $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0,\varnothing)F(\underline{1} \supset \varphi)\}$. Hence, by soundness (Proposition 4.11), φ must be valid.

For the converse implication, suppose ISVALID(φ) does not return **true**. Since the procedure terminates, the while loop performed by CONSTRUCTTABLEAU($F(\underline{1} \supset \varphi)$) ends after k iterations for some $k \in \mathbb{N}$, and CONSTRUCTTABLEAU($F(\underline{1} \supset \varphi)$) returns $(\mathscr{T}_k, U_k)$. But since ISVALID(φ) returns **false**, $(\mathscr{T}_k, U_k)$ is not closed. Thus, $(\mathscr{T}_k, U_k)$ contains an open branch $\mathscr{B}$ and each node in $\mathscr{B}$ is marked as finished. Note that $\mathscr{B}$ being open implies that $\mathscr{B}_i$ is open for each $1 \leq i \leq k$.
We claim that each condition of Definition 4.13 holds for $U(\mathscr{B})$ (i.e., that $U(\mathscr{B})$ is downward saturated). This should not be surprising, since the applications of rules to $\mathscr{B}_i$ for each iteration i of the while loop in CONSTRUCTTABLEAU are essentially guided by the aim of ensuring that this claim holds.

1. Suppose $(w, v, \underline{t}) \in cons(U(\mathscr{B}))$ for some $w, v \in \Sigma$ and $t \in H$. $cons(U(\mathscr{B}))$ starts out empty, and only grows through some applications of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ to the labeled tree (see lines 74 to 88). In particular, during some iteration $1 \leq i \leq k$, an unfinished node n in $\mathscr{B}_{i-1}$ was picked where $w = world(U(n)) \in worlds(U(\mathscr{B}_{i-1})) \subseteq worlds(U(\mathscr{B}))$, and $(w, v, \underline{t})$ was added to $cons(U(\mathscr{B}_i))$ through an application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$, where $v \in worlds(U(\mathscr{P}_i)) \subseteq worlds(U(\mathscr{B}_i)) \subseteq worlds(U(\mathscr{B}))$.

   We now need to show that $R_{U(\mathscr{B})}$ is a partial function from $worlds(U(\mathscr{B}))^2$ to H. Suppose $((w, v), t), ((w, v'), t') \in R_{U(\mathscr{B})}$. Recalling Definition 4.12, this implies that $(w, v, \underline{t}), (w, v', \underline{t'}) \in cons(U(\mathscr{B}))$.
   Now, there exists a minimal $1 \leq i \leq k$ such that $(w, v, \underline{t}) \in cons(U(\mathscr{P}_i))$. Similarly, there exists a minimal $1 \leq j \leq k$ such that $(w, v', \underline{t'}) \in cons(U(\mathscr{P}_j))$.
   Iterations i, j must involve applications of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ to introduce $(w, v, \underline{t})$ and $(w, v', \underline{t})$ respectively.
   Suppose $i \neq j$. Without loss of generality, let us assume $i < j$. Then, in iteration j, the application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ to $\mathscr{B}_{j-1}$ ensures that $v \neq v'$ (see line 75 or 82).
   So now suppose $v = v'$. Then we must have $i = j$. But only one application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ is applied to $\mathscr{B}_{i-1}$ during iteration i, and we must have $(w, v, \underline{t}) = (w, v', \underline{t'})$. Hence, $t = t'$, so Condition (1) holds for $U(\mathscr{B})$.

2. If $U(\mathscr{B})$ is an instantiation of the numerator of a rule $\rho \in \{p\bot_1, p\bot_2, p\bot_3, p\bot_4, p\bot_5\}$, then $(w, \varnothing)\bot \in U(\mathscr{B})$ for some $w \in \Sigma$ (see lines 13 to 15). But this cannot be the case, as $\mathscr{B}$ is open. Thus, Condition (2) must hold for $U(\mathscr{B})$.

3. Suppose $(w, \sigma)T(\underline{a} \supset (\varphi \wedge \psi)) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 0$. So, for some node n in $\mathscr{B}$, $U(n) = (w, \sigma)T(\underline{a} \supset (\varphi \wedge \psi))$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. In this iteration, for $\mathscr{B}_{i-1}$ (which recall is open), the steps in lines 37 and 38 ensure that $(w, \sigma')T(\underline{a} \supset \varphi), (w, \sigma')T(\underline{a} \supset \psi) \in U(\mathscr{P}_i) \subseteq U(\mathscr{B})$ where $\sigma' = cons(U(\mathscr{B}_{i-1})) \subseteq \Sigma^2 \times \underline{H}$. Thus, Condition (3) holds for $U(\mathscr{B})$.

4. Suppose $(w, \sigma)F(\underline{a} \supset (\varphi \wedge \psi)) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value $a \neq 0$. So, for some node n in $\mathscr{B}$, $U(n) = (w, \sigma)F(\underline{a} \supset (\varphi \wedge \psi))$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. In this iteration, the steps in lines 40 and 41 performed for $\mathscr{B}_{i-1}$ amount to splitting $\mathscr{B}_{i-1}$ along two paths. One path contains a node labelled with $(w, \sigma')F(\underline{a} \supset \varphi)$ and the other path contains a node labelled $(w, \sigma')F(\underline{a} \supset \psi)$, where $\sigma' = cons(U(\mathscr{B}_{i-1})) \subseteq \Sigma^2 \times \underline{H}$. $\mathscr{P}_i$ is one of these paths, and hence $(w, \sigma')F(\underline{a} \supset \varphi)$ or $(w, \sigma')F(\underline{a} \supset \varphi)$ is in $U(\mathscr{P}_i) \subseteq U(\mathscr{B})$. Thus, Condition (4) holds for $U(\mathscr{B})$.

5. To show Condition (5) holds, we employ an argument similar to that in 3.

6. To show Condition (6) holds, we employ an argument similar to that in 4.

7. Suppose $(w, \sigma)F(\underline{a} \supset (\varphi \supset \psi)) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a. So, for some node n in $\mathscr{B}$, $U(n) = (w, \sigma)F(\underline{a} \supset (\varphi \supset \psi))$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. Let $A \coloneqq \{t \in H \mid t \leq a \text{ and } t \neq 0\}$. Then, the steps in lines 49 to 52 extend $\mathscr{B}_{i-1}$ along $|A|$ paths $\{\mathscr{P}^t\}_{t \in A}$, where for each $t \in A$, a node labelled with $(w, \sigma')T(\underline{t} \supset \varphi)$ and a node labelled with $(w, \sigma')F(\underline{t} \supset \psi)$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$ are in $\mathscr{P}^t$. But, for some $t \in A$, $\mathscr{P}_i$ must be $\mathscr{P}^t$. Hence, $(w, \sigma')T(\underline{t} \supset \varphi), (w, \sigma')F(\underline{t} \supset \psi) \in U(\mathscr{P}_i) \subseteq U(\mathscr{B})$ and so, Condition (7) holds for $U(\mathscr{B})$.

8. Suppose $(w,\sigma)T(\underline{a} \supset (\varphi \supset \psi)) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a. So, for some node n in $\mathscr{B}$, $U(n) = (w,\sigma)T(\underline{a} \supset (\varphi \supset \psi))$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. Let $A := \{t \in H \mid t \leq a \text{ and } t \neq 0\}$. Consider a minimal set satisfying the property that for each $t \in A$, the set either contains $(w, cons(\mathscr{B}_{i-1}))F(\underline{t} \supset \varphi)$ or it contains $(w, cons(\mathscr{B}_{i-1}))T(\underline{t} \supset \psi)$. There are $2^{|A|}$ such sets, let us denote them P_q for $1 \leq q \leq 2^{|A|}$. Then, the steps in lines 54 to 63 extend $\mathscr{B}_{i-1}$ along $2^{|A|}$ paths $\{\mathscr{P}^q\}_{q \in \{1 \ldots 2^{|A|}\}}$, where for each $q \in \{1 \ldots 2^{|A|}\}$, $U(\mathscr{P}^q) = P_q$. That is, each $\mathscr{P}^q$ contains a node labelled with $(w, cons(\mathscr{B}_{i-1}))F(\underline{t_1} \supset \varphi)$ or a node labelled with $(w, cons(\mathscr{B}_{i-1}))T(\underline{t_1} \supset \psi)$ for each $t \in A$. But, for some $1 \leq q \leq 2^{|A|}$ we have that $\mathscr{P}_i$ is the same as $\mathscr{P}^q$, and $U(\mathscr{P}_i) \subseteq U(\mathscr{B})$. So, Condition (8) holds for $U(\mathscr{B})$.

9. Suppose $(w,\sigma)T(\underline{a} \supset \Box\varphi) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a. So, for some node n in $\mathscr{B}$, $U(n) = (w,\sigma)T(\underline{a} \supset \Box\varphi)$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. Let $v \in \Sigma$, $t \in H$ and suppose $(w, v, \underline{t}) \in cons(U(\mathscr{B}))$. There exists a minimal $1 \leq j \leq k$ such that $(w, v, \underline{t}) \in cons(U(\mathscr{P}_j))$. We have two cases.

   - $j < i$. Then $(w, v, \underline{t}) \in cons(U(\mathscr{B}_j)) \subseteq cons(U(\mathscr{B}_{i-1}))$, and the steps in lines 64 to 68 performed for $\mathscr{B}_{i-1}$ ensure that $(v, \sigma')T(\underline{a \wedge t} \supset \varphi) \in U(\mathscr{B})$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.
   - $j \geq i$. Then n has already been marked as finished by the time we get to iteration j. Further, iteration j must involve an application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ for $\mathscr{B}_{j-1}$, and so the call to REACTIVATE for $\mathscr{B}_{j-1}$ ensures that $(v, \sigma' \cup \{(w, v, \underline{t})\})T(\underline{a \wedge t} \supset \varphi) \in U(\mathscr{B})$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

   In either case, $(v, \sigma'')T(\underline{a \wedge t} \supset \varphi) \in U(\mathscr{B})$ for some $\sigma'' \subseteq \Sigma^2 \times \underline{H}$. So, Condition (9) holds for $U(\mathscr{B})$.

10. To show Condition (10) holds, we employ an argument similar to that in 9.

11. Suppose $(w,\sigma)F(\underline{a} \supset \Box\varphi) \in U(\mathscr{B})$ for some $w \in \Sigma$, $\sigma \subseteq \Sigma^2 \times \underline{H}$ and truth value a. So, for some node n in $\mathscr{B}$, $U(n) = (w,\sigma)F(\underline{a} \supset \Box\varphi)$. Since each node in $\mathscr{B}$ is marked as finished, n must have been picked during some iteration $1 \leq i \leq k$. Let $A := \{t \in H \mid a \wedge t \neq 0\}$. Then, the steps in lines 75 to 79 pick some $v \in \Sigma$ and extend $\mathscr{B}_{i-1}$ along $|A|$ paths $\{\mathscr{P}^t\}_{t \in A}$, where for each $t \in A$, a node labelled with $(v, \sigma' \cup \{(w, v, \underline{t})\})F(\underline{a \wedge t} \supset \varphi)$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$ is in $\mathscr{P}^t$. But, for some $t \in A$, $\mathscr{P}_i$ is the same as $\mathscr{P}^t$. Hence, $(w, v, \underline{t}) \in cons(U(\mathscr{P}_i)) \subseteq cons(U(\mathscr{B}))$ and $(v, \sigma'')F(\underline{a \wedge t_i} \supset \varphi) \in U(\mathscr{P}_i) \subseteq U(\mathscr{B})$ for some $\sigma'' \subseteq \Sigma^2 \times \underline{H}$. So, Condition (11) holds for $U(\mathscr{B})$.

12. To show Condition (12) holds, we employ an argument similar to that in 11.

13. The steps in lines 17 to 20 ensure Condition (13).

14. The steps in lines 22 to 25 ensure Condition (14).

15. The steps in lines 27 to 30 ensure Condition (15).

16. The steps in lines 32 to 35 ensure Condition (16).

Thus, $U(\mathscr{B})$ is downward saturated. Then, by Lemma 4.15, $U(\mathscr{B})$ is satisfiable. But $(w_0, \varnothing)F(\underline{1} \supset \varphi) \in U(\mathscr{B})$, and hence φ cannot be valid. □

With the above result in hand, we are in a position to prove completeness.

Corollary 4.24. *$p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ is (weakly) complete with respect to the class of all $\mathcal{H}$-frames.*

Proof. We prove the contrapositive. Suppose $\nvdash_{p\mathcal{C}\mathbf{K}^{\mathcal{H}}} \varphi$. That is, taking any $w \in \Sigma$, there does not exist a closed $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $(w, \varnothing)F(\underline{1} \supset \varphi)$. By Lemma 4.22, CONSTRUCTTABLEAU($F(\underline{1} \supset \varphi)$) returns the labelled tree $(\mathscr{T}, U)$, where $T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0, \sigma)F(\underline{1} \supset \varphi)\}$. This implies that ISVALID(φ) cannot possibly return **true**, as such an eventuality relies on $(\mathscr{T}, U)$ being closed, which would imply that $T_{(\mathscr{T},U)}$ is a closed $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $(w_0, \varnothing)F(\underline{1} \supset \varphi)$. Thus, by Proposition 4.23, we can conclude that φ is not valid. □

Corollary 4.25. ISVALID *is a decision procedure for the logic $\mathbf{K}^{\mathcal{H}}$.*

Proof. This simply follows from form Proposition 4.20 and 4.23. □

A concrete implementation has been written in python and is publicly available. For those interested, see the GitHub repository (https://github.com/WeAreDevo/Many-Valued-Modal-Tableau).

The decision procedure also suggests a finite frame property, which we present now. Let us say that an $\mathcal{H}$-frame $\mathfrak{F} = (W, R)$ is finite iff the set of worlds W is finite. A class of $\mathcal{H}$-frames $\mathcal{F}$ is of ***finite character*** iff each $\mathcal{H}$-frame in $\mathcal{F}$ is finite.

Definition 4.26. Let $\Lambda \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$. Λ is said to have the ***finite frame property*** iff $\Lambda = \Lambda_{\mathcal{F}}$ for some class of frames $\mathcal{F}$ of finite character.

Corollary 4.27. *$\mathbf{K}^{\mathcal{H}}$ has the finite frame property.*

Proof. Consider the class $\mathcal{F}$ of all *finite* $\mathcal{H}$-frames. We claim that $\mathbf{K}^{\mathcal{H}} = \Lambda_{\mathcal{F}}$.
Clearly $\mathbf{K}^{\mathcal{H}} \subseteq \Lambda_{\mathcal{F}}$ (since $\mathcal{F}$ is a subclass of the class of all $\mathcal{H}$-frames). To show $\Lambda_{\mathcal{F}} \subseteq \mathbf{K}^{\mathcal{H}}$, consider a formula $\varphi \notin \mathbf{K}^{\mathcal{H}}$. We argue that $\varphi \notin \Lambda_{\mathcal{F}}$.
Since $\varphi \notin \mathbf{K}^{\mathcal{H}}$, φ is not valid. So, as in the second part of the proof for Proposition 4.23, CONSTRUCTTABLEAU($F(\underline{1} \supset \varphi)$) returns a labeled tree containing an open branch $\mathscr{B}$, where $U(\mathscr{B})$ is downward saturated. $U(\mathscr{B})$ induces an $\mathcal{H}$-model $\mathfrak{M}_{U(\mathscr{B})}$ which is a counter model for φ. But $\mathfrak{M}_{U(\mathscr{B})}$ is based on an $\mathcal{H}$-frame (W, R) where $W = worlds(U(\mathscr{B}))$ (Recall the proof of Lemma 4.15). Clearly, the only members of $worlds(U(\mathscr{B}))$ are the initial world w_0, along with a distinct world v introduced by each application of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$. But the number of applications of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ is bounded above by a finite function of $Mdegree(\varphi)$ and $|H|$. Hence $worlds(U(\mathscr{B}))$ is finite. And since $\mathfrak{M}_{U(\mathscr{B})}$ is a counter model for φ, we must have $\varphi \notin \Lambda_{\mathcal{F}}$. □

CONSTRUCTTABLEAU may be adapted to define various other decision procedures deciding problems regarding our many-valued logics. That is the content of the following remarks.

Remark 4.28. By definition, decision procedures are finitary manipulations of finite objects. As such, we could not decide the full problem of entailment, which may require the input of an infinite set of formulas. However, we could still decide the problem of *finite* entailment. That is, suppose we are given a finite set of formulas $\Gamma \cup \{\varphi\}$, and we wish to determine if Γ entails φ. Firstly, note that as it stands, CONSTRUCTTABLEAU takes in a single element of SBI. However, there is no technical reason why we could not take in any finite subset of SBI. Indeed, this will be useful in the current circumstance. Let us assume CONSTRUCTTABLEAU requires a finite set $X \subseteq SBI$. And that, instead of initializing $(\mathscr{T}, U)$ as a single node labeled tree, we initialize $(\mathscr{T}, U)$ as a single *branch* labeled tree, where $U(\mathscr{B}_0) = \{(w_0, \varnothing)x \mid x \in X\}$. Subsequently, assuming we let $X = \{T(\underline{1} \supset \gamma) \mid \gamma \in \Gamma\} \cup \{F(\underline{1} \supset \varphi)\}$, if CONSTRUCTTABLEAU($X$) returns a closed labeled tree, we can conclude that Γ does entail φ. If, on the other hand, $\textsc{constructTableau}(X)$ returns a labeled tree containing an open branch $\mathscr{B}$, then $U(\mathscr{B})$ is downward saturated and induces a model in which, for a particular world, each member of X is satisfied at that world. And this would mean that Γ does not entail φ.

Remark 4.29. In this dissertation, we have only been concerned with validity defined in terms of the $\mathcal{H}$-truth value 1. However, the presence of $\mathcal{H}$-truth value constants and the propertis of Heyting algebras makes this notion fairly general. For instance, let a be an arbitrary $\mathcal{H}$-truth value and suppose we are interested in the 'validity-type' problem of whether a formula φ has a value of at least a at every world in every $\mathcal{H}$-model. We can simply reduce this to the problem of the validity of the formula $\underline{a} \supset \varphi$.
Now, let us briefly touch on the satisfiability problem. There are various different 'satisfiability-type' notions we could consider. For instance, we may say that a formula φ is satisfiable iff $V(\mathfrak{s}, \varphi) \neq 0$ for some $\mathcal{H}$-model $((W, R), V)$ and $\mathfrak{s} \in W$. In this case, satisfiability is dual to validity. In other words, φ is satisfiable iff $\neg\varphi$ is not valid.
Alternatively, we may wish to fix some $a \in H$ and say that φ is satisfiable iff $V(\mathfrak{s}, \varphi) \geq a$ for some $\mathcal{H}$-model $((W, R), V)$ and $\mathfrak{s} \in W$. No longer is satisfiability dual to validity. Nonetheless, it is simple to use $\textsc{constructTableau}$ for deciding this notion of satisfiability. In particular, we would execute $\textsc{constructTableau}(T(\underline{a} \supset \varphi))$. If we get a closed labeled tree, then φ is not satisfiable. Else, φ is satisfiable. It is satisfied by a model induced by the labels in an open branch of the constructed tree.

Predictable changes to $\textsc{constructTableau}$ would allow us to define desicion procedures for the logics studied in Section 3[11]. We will not go into details in this dissertation. Instead, we consider the logic $\mathbf{KB}^{\mathcal{H}}_d$ next.

4.3 Tableau System for $\mathbf{KB}^{\mathcal{H}}_d$

Obtaining a destructive unprefixed tableau system for $\mathbf{KB}^{\mathcal{H}}_d$ for all $d \in H$ was not possible in general. In this subsection, we briefly consider simple modifications of the rules $p\mathbf{K}F\Box$ and $p\mathbf{K}F\Diamond$, from which we obtain a prefixed tableau system for $\mathbf{KB}^{\mathcal{H}}_d$ for all $d \in H$.
Let us fix an arbitrary $d \in H$. The next page presents the rules $p\mathbf{KB}F\Box_d$ and $p\mathbf{KB}F\Diamond_d$. We proceed to argue that the tableau system

$$\begin{aligned} p\mathcal{C}\mathbf{KB}^{\mathcal{H}}_d := \{ & p\bot_1, p\bot_2, p\bot_3, p\bot_4, p\bot_5, pF{\geq}, pT{\geq}, pF{\leq}, pT{\leq}, pT{\wedge}, pF{\wedge}, pT{\vee}, pF{\vee}, \\ & pT{\supset}, pF{\supset}, p\mathbf{K}T\Box, p\mathbf{K}T\Diamond, p\mathbf{KB}F\Box_d, p\mathbf{KB}F\Diamond_d \} \end{aligned}$$

is sound and complete with respect to $\mathsf{Symm}^{\mathcal{H}}_d$.

[11]However, the case of $\mathbf{K4}^{\mathcal{H}}_d$ requires a quite different approach to proving termination.

$(p\mathbf{KB}F\Box_d)$

$$X; (w,\sigma)F(\underline{a} \supset \Box\varphi)$$

$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_1}), (v, w, \underline{t_1^1})\})$ $F(\underline{a \wedge t_1} \supset \varphi)$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_1}), (v, w, \underline{t_1^{k_1}})\})$ $F(\underline{a \wedge t_1} \supset \varphi)$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_n}), (v, w, \underline{t_n^1})\})$ $F(\underline{a \wedge t_n} \supset \varphi)$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_n}), (v, w, \underline{t_n^{k_n}})\})$ $F(\underline{a \wedge t_n} \supset \varphi)$

Where v is any symbol of Σ that is not in $worlds(\mathcal{N})$, $t_1, \ldots, t_n$ are all the $\mathcal{H}$-truth values s.t. $a \wedge t_i \neq 0$, and for each $i \in \{1, \ldots, n\}$, $\{t_i^1, \ldots, t_i^{k_i}\} = \{t \in H \mid d \wedge t_i = d \wedge t\}$.

$(p\mathbf{KB}F\Diamond_d)$

$$X; (w,\sigma)F(\Diamond\varphi \supset \underline{a})$$

$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_1}), (v, w, \underline{t_1^1})\})$ $F(\varphi \supset \underline{t_1 \Rightarrow a})$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_1}), (v, w, \underline{t_1^{k_1}})\})$ $F(\varphi \supset \underline{t_1 \Rightarrow a})$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_n}), (v, w, \underline{t_n^1})\})$ $F(\varphi \supset \underline{t_n \Rightarrow a})$	$\ldots$	$\mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_n}), (v, w, \underline{t_n^{k_n}})\})$ $F(\varphi \supset \underline{t_n \Rightarrow a})$

Where v is any symbol of Σ that is not in $worlds(\mathcal{N})$, $t_1, \ldots, t_n$ are all the $\mathcal{H}$-truth values s.t. $t_i \Rightarrow a \neq 1$, and for each $i \in \{1, \ldots, n\}$, $\{t_i^1, \ldots, t_i^{k_i}\} = \{t \in H \mid d \wedge t_i = d \wedge t\}$.

Remark 4.30. For $d = 1$, $\{t \in H \mid d \wedge t_i = d \wedge t\} = \{t_i\}$.

Proposition 4.31. *$p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ is sound with respect to $\mathsf{Symm}_d^{\mathcal{H}}$.*

Proof. It suffices to show that each rule in $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ preserves $\mathsf{Symm}_d^{\mathcal{H}}$-satisfiability. Let $\rho \in p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ and suppose that the numerator $\mathcal{N}$ of ρ is $\mathsf{Symm}_d^{\mathcal{H}}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M} = ((W, R), V)$ based on a frame from $\mathsf{Symm}_d^{\mathcal{H}}$, and an interpretation I of $\mathcal{N}$ in $\mathfrak{M}$ s.t. $\mathcal{N}$ is satisfied under I. We wish to show that at least one of the denominators $\mathcal{D}$ is $\mathsf{Symm}_d^{\mathcal{H}}$-satisfiable. We only need to consider the case in which $\rho = p\mathbf{KB}F\Box_d$ or $\rho = p\mathbf{KB}F\Diamond_d$. The other cases follow from Lemma 4.9, with $\mathcal{F} = \mathsf{Symm}_d^{\mathcal{H}}$.

- $\rho = p\mathbf{KB}F\Box_d$.

$$\mathcal{N} = X; (w, \sigma)F(\underline{a} \supset \Box\varphi)$$

So $F(\underline{a} \supset \Box\varphi)$ is satisfied by $\mathfrak{M}$ at $I(w)$. That is, $V(I(w), \underline{a} \supset \Box\varphi) \neq 1$. Or equivalently,

$$\begin{aligned} a &\nleq V(I(w), \Box\varphi) \\ &= \bigwedge\{R(I(w), \mathfrak{s}) \Rightarrow V(\mathfrak{s}, \varphi) \mid \mathfrak{s} \in W\}. \end{aligned}$$

Thus, for some $\mathfrak{s} \in W$, we have $a \nleq R(I(w), \mathfrak{s}) \Rightarrow V(\mathfrak{s}, \varphi)$. Or equivalently,

$$a \wedge R(I(w), \mathfrak{s}) \nleq V(\mathfrak{s}, \varphi). \tag{4.31.1}$$

Suppose $R(I(w), \mathfrak{s}) = t_i \in H$. Clearly (4.31.1) implies $a \wedge t_i \neq 0$. Suppose $R(\mathfrak{s}, I(w)) = t \in H$. Since $(W, R) \in \mathsf{Symm}_d^{\mathcal{H}}$, we have $d \wedge t_i = d \wedge t$. Let $v \in \Sigma$ be any symbol that is not already in $worlds(\mathcal{N})$. We extend the interpretation I to v. Specifically, consider $I' := I \cup \{(v, \mathfrak{s})\}$. I' is an interpretation of

$$\mathcal{D} = \mathcal{N}; (v, \sigma' \cup \{(w, v, \underline{t_i}), (v, w, \underline{t})\})F(\underline{a \wedge t_i} \supset \varphi)$$

in $\mathfrak{M}$.
And by (4.31.1), we have that $\mathcal{D}$ is satisfied under I'.

- $\rho = p\mathbf{KB}F\Diamond_d$.
The argument is similar to the preceding case.

∎

Proposition 4.32. *$p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ is (weakly) complete with respect to $\mathsf{Symm}_d^{\mathcal{H}}$.*

Proof. Let us introduce the notion of $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturation. Say that $S \subseteq pSBI$ is downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturated iff S is downward saturated (Definition 4.13), and

1'. For all $w, v \in \Sigma$, $t \in H$, if $(w, v, \underline{t}) \in cons(S)$, then $(v, w, \underline{t'}) \in cons(S)$ for some $t' \in \mathcal{H}$ such that $t \wedge d = t' \wedge d$.

If S is downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturated, we may use the same approach as in Lemma 4.15 to construct/induce an $\mathcal{H}$-model $\mathfrak{M}_S$ and an interpretation I of S in $\mathfrak{M}_S$ s.t. S is satisfied under I. In addition, since S satisfies (1'), it is clear that the model $\mathfrak{M}_S$ we construct is in fact based on a frame from $\mathsf{Symm}_d^{\mathcal{H}}$. Hence, S is $\mathsf{Symm}_d^{\mathcal{H}}$-satisfiable whenever S is downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturated.

We can modify CONSTRUCTTABLEAU such that ISVALID is a decision procedure for $\mathbf{KB}_d^{\mathcal{H}}$. Specifically, we just replace lines 74 to 88 with the following:

72: **if** β is of the form $F(\underline{a} \supset \Box\varphi)$ **then**
73: Pick some $v \in \Sigma$ that does not already occur in $worlds(U(\mathscr{B}))$.
74: **for** each $t \in H$ such that $a \wedge t \neq 0$ **do**
75: **for** each $t' \in H$ for which $d \wedge t' = d \wedge t$ **do**
76: Create a new node n' with
$U(n') = (v, \sigma' \cup \{(w, v, \underline{t}), (v, w, \underline{t'})\})F(\underline{a \wedge t} \supset \varphi)$
77: Add n' as a child of l
78: **end for**
79: **end for**
80: REACTIVATE(n)
81: **else if** β is of the form $F(\Diamond\varphi \supset \underline{a})$ **then**
82: Pick some $v \in \Sigma$ that does not already occur in $worlds(U(\mathscr{B}))$.
83: **for** each $t \in H$ such that $t \Rightarrow a \neq 1$ **do**
84: **for** each $t' \in H$ for which $d \wedge t' = d \wedge t$ **do**
85: Create a new node n' with
$U(n') = (v, \sigma' \cup \{(w, v, \underline{t}), (v, w, \underline{t'})\})F(\varphi \supset \underline{t \Rightarrow a})$
86: Add n' as a child of l
87: **end for**
88: **end for**
89: REACTIVATE(n)
90: **end if**

By the same argument given for Proposition 4.20, with slight modifications to the base case proof of 4.20.2, we can show that ISVALID(φ) still terminates for every formula φ.

Further, most of the argument for Lemma 4.22 still applies. Only now, the updated steps in the procedure amount to replacing applications of $p\mathbf{K}F\Box$ and $p\mathbf{K}F\Diamond$ with applications of $p\mathbf{KB}F\Box_d$ and $p\mathbf{KB}F\Diamond_d$ respectively. And as such, the labeled tree constructed by CONSTRUCTTABLEAU(α) corresponds to a $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-tableau for $\{(w_0, \sigma)\alpha\}$.

Finally, an argument essentially the same as that for Proposition 4.23 can be used to show that ISVALID(φ) returns **true** iff φ is $\mathsf{Symm}_d^{\mathcal{H}}$-valid. Note, in particular, that the modified steps in CONSTRUCTTABLEAU given above still ensure that Conditions (9),(10),(11) and (12) hold for the set of labels of an open branch in the constructed labeled tree. The major difference arises in a need to show not only downward saturation, but downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturation. Suppose the returned labeled tree $(\mathscr{T}, U)$ contains an open branch $\mathscr{B}$, where each node in $\mathscr{B}$ is marked as finished. We require that Condition (1') introduced above holds for $U(\mathscr{B})$. But this can be seen to inherently follow from the way we have altered the steps in CONSTRUCTTABLEAU. Every time we introduce a constraint into $\mathscr{B}$, we ensure an appropriate reversed constraint accompanies it.

$\Box$

Note that for the updated decision procedure, the number of applications of $p\mathbf{K}F\Box$ or $p\mathbf{K}F\Diamond$ is still bounded above by a finite function of $Mdegree(\varphi)$ and $|H|$. Thus, we get the following result.

Corollary 4.33. $\boldsymbol{KB}_d^{\mathcal{H}}$ *has the finite frame property.*

In particular, $\mathbf{KB}_d^{\mathcal{H}} = \Lambda_{\mathcal{F}}$ where $\mathcal{F}$ is the class of finite members of $\mathsf{Symm}_d^{\mathcal{H}}$.

Example 4.34. Consider $\varphi \equiv p \supset \Box\Diamond p$. φ is $\mathsf{Symm}_1^{\mathcal{H}^3}$-valid. A semantic argument is as follows: Let $\mathfrak{M} = ((W, R), V)$ be any $\mathcal{H}$-model such that $(W, R) \in \mathsf{Symm}_1^{\mathcal{H}}$. For an arbitrary $\mathfrak{s} \in W$, we have

$$\begin{aligned} V(\mathfrak{s}, \varphi) &= V(\mathfrak{s}, p) \Rightarrow \bigwedge \left\{ R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \bigvee \{R(\mathfrak{v}, \mathfrak{r}) \wedge V(\mathfrak{r}, p) \mid \mathfrak{r} \in W\} \mid \mathfrak{v} \in W \right\} \\ &\geq V(\mathfrak{s}, p) \Rightarrow \bigwedge \left\{ \bigvee \{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \big(R(\mathfrak{v}, \mathfrak{r}) \wedge V(\mathfrak{r}, p)\big) \mid \mathfrak{r} \in W\} \mid \mathfrak{v} \in W \right\} \quad (2.24.15, 2.17.4, 2.24.5). \end{aligned}$$

But taking $\mathfrak{r} = \mathfrak{s}$, we have

$$\bigvee \{R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \big(R(\mathfrak{v}, \mathfrak{r}) \wedge V(\mathfrak{r}, p)\big) \mid \mathfrak{r} \in W\} \geq R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \big(R(\mathfrak{v}, \mathfrak{s}) \wedge V(\mathfrak{s}, p)\big)$$

And $R(\mathfrak{s}, \mathfrak{v}) = R(\mathfrak{v}, \mathfrak{s})$ since $(W, R) \in \mathsf{Symm}_1^{\mathcal{H}}$. Thus,

$$\begin{aligned} V(\mathfrak{s}, \varphi) &\geq V(\mathfrak{s}, p) \Rightarrow \bigwedge \left\{ R(\mathfrak{s}, \mathfrak{v}) \Rightarrow \big(R(\mathfrak{v}, \mathfrak{s}) \wedge V(\mathfrak{s}, p)\big) \mid \mathfrak{v} \in W \right\} \\ &= \bigwedge \left\{ V(\mathfrak{s}, p) \Rightarrow \big((R(\mathfrak{s}, \mathfrak{v}) \Rightarrow R(\mathfrak{v}, \mathfrak{s})) \wedge (R(\mathfrak{s}, \mathfrak{v}) \Rightarrow V(\mathfrak{s}, p))\big) \mid \mathfrak{v} \in W \right\} \\ &= \bigwedge \left\{ (V(\mathfrak{s}, p) \wedge R(\mathfrak{s}, \mathfrak{v})) \Rightarrow V(\mathfrak{s}, p) \mid \mathfrak{v} \in W \right\} \\ &= 1 \end{aligned}$$

This is confirmed by executing $\textsc{constructTableau}(F(1 \supset \varphi))$ with d set to 1, which produces the labeled tree in Figure 4.4 (assuming unfinished nodes are picked in a first in first out fashion). This labeled tree corresponds to a $p\mathcal{C}\mathbf{KB}_1^{\mathcal{H}^3}$-proof for φ.

In contrast, φ is not $\mathsf{Symm}_h^{\mathcal{H}^3}$-valid. With d set to h, $\textsc{constructTableau}(F(1 \supset \varphi))$ produces the labeled tree in Figure 4.6, which corresponds to an open $p\mathcal{C}\mathbf{KB}_h^{\mathcal{H}^3}$-tableau. The branches with ellipses are the same as the branches in Figure 4.4, and hence close. Let us consider the open branch $\mathscr{B}$ of the tree. $U(\mathscr{B})$ is downward $p\mathcal{C}\mathbf{KB}_h^{\mathcal{H}^3}$-saturated, and induces the following $\mathcal{H}^3$-model based on a frame from $\mathsf{Symm}_h^{\mathcal{H}^3}$:

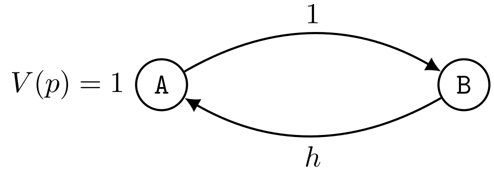


Figure 4.5

As the reader can confirm, the value of φ at A is h. And so, this model is in fact a countermodel for φ.

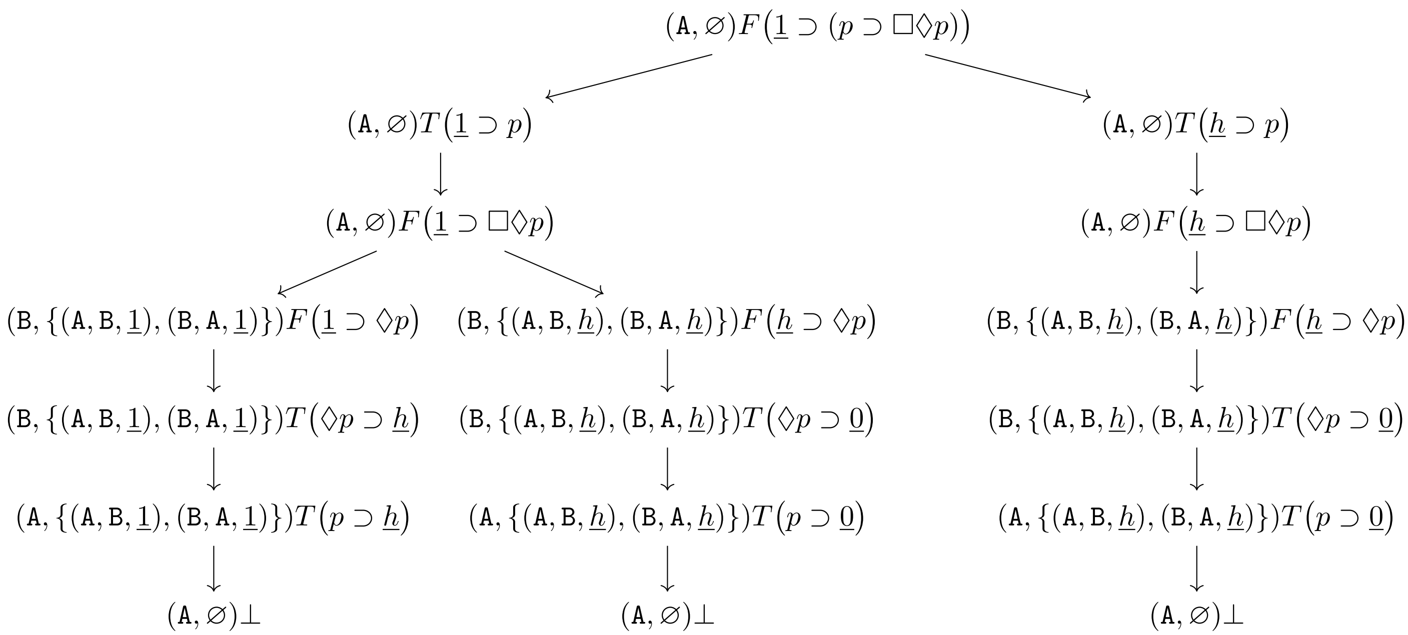

$(\mathtt{A}, \varnothing) F\big(\underline{1} \supset (p \supset \Box\Diamond p)\big)$
$(\mathtt{A}, \varnothing) T\big(\underline{1} \supset p\big)$
$(\mathtt{A}, \varnothing) T\big(\underline{h} \supset p\big)$
$(\mathtt{A}, \varnothing) F\big(\underline{1} \supset \Box\Diamond p\big)$
$(\mathtt{A}, \varnothing) F\big(\underline{h} \supset \Box\Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{1})\}) F\big(\underline{1} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) F\big(\underline{h} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) F\big(\underline{h} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{1})\}) T\big(\Diamond p \supset \underline{h}\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) T\big(\Diamond p \supset \underline{0}\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) T\big(\Diamond p \supset \underline{0}\big)$
$(\mathtt{A}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{1})\}) T\big(p \supset \underline{h}\big)$
$(\mathtt{A}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) T\big(p \supset \underline{0}\big)$
$(\mathtt{A}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\}) T\big(p \supset \underline{0}\big)$
$(\mathtt{A}, \varnothing)\bot$
$(\mathtt{A}, \varnothing)\bot$
$(\mathtt{A}, \varnothing)\bot$

Figure 4.4

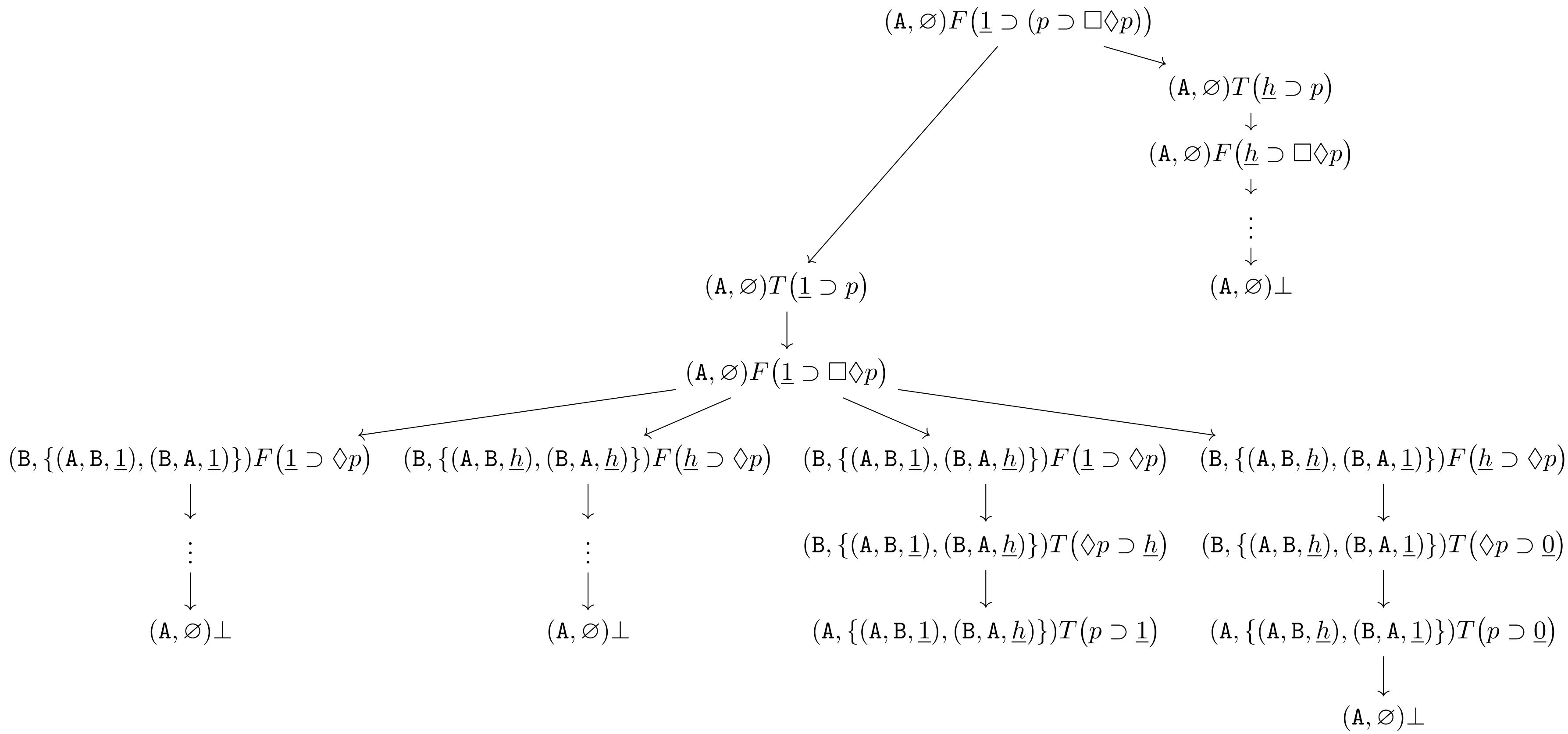

$(\mathtt{A}, \varnothing)F\big(\underline{1} \supset (p \supset \Box\Diamond p)\big)$
$(\mathtt{A}, \varnothing)T\big(\underline{h} \supset p\big)$
$(\mathtt{A}, \varnothing)F\big(\underline{h} \supset \Box\Diamond p\big)$
$(\mathtt{A}, \varnothing)\bot$
$(\mathtt{A}, \varnothing)T\big(\underline{1} \supset p\big)$
$(\mathtt{A}, \varnothing)F\big(\underline{1} \supset \Box\Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{1})\})F\big(\underline{1} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{h})\})F\big(\underline{h} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{h})\})F\big(\underline{1} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{1})\})F\big(\underline{h} \supset \Diamond p\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{h})\})T\big(\Diamond p \supset \underline{h}\big)$
$(\mathtt{B}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{1})\})T\big(\Diamond p \supset \underline{0}\big)$
$(\mathtt{A}, \varnothing)\bot$
$(\mathtt{A}, \varnothing)\bot$
$(\mathtt{A}, \{(\mathtt{A}, \mathtt{B}, \underline{1}), (\mathtt{B}, \mathtt{A}, \underline{h})\})T\big(p \supset \underline{1}\big)$
$(\mathtt{A}, \{(\mathtt{A}, \mathtt{B}, \underline{h}), (\mathtt{B}, \mathtt{A}, \underline{1})\})T\big(p \supset \underline{0}\big)$
$(\mathtt{A}, \varnothing)\bot$

Figure 4.6

Remark 4.35. At this point, it is natural to ask what the decision procedures tell us about the computational complexity of these logics. A precedent for this kind of research has been set in [7, 11]. Here we only give rudimentary thoughts on the problem, and leave a more thorough study for future work. Firstly, note that one could imagine a recursive implementation of constructTableau, in which we only consider a single branch at a time. Such a branch has length that is bounded above by a polynomial function of $|\varphi| \times |H|$. Thus, the logics decided by our decision procedure are in the class $\mathbf{PSPACE}(|\varphi| \times |H|)$. In the case where $\mathcal{H} = \mathbf{2}$ (and hence $|H|$ is constant), we see that our decision procedure has a complexity coinciding with the **PSPACE** lower bound of standard modal logic.

5 Conclusion

This concludes our modest contribution to the proof theory of Fitting's many-valued modal logic. It is hoped that the reader has been convinced of the virtues of using tableaux in this setting. Various properties that are desirable in any proof sysyem are pinpointed in [40].
The main contributions of this dissertation were given in Sections 3 and 4.

In section 3, we have expanded upon the tableau system introduced by Fitting in [24]. This resulted in the creation of unprefixed tableau systems for the logics $\mathbf{K4}_d^{\mathcal{H}}$ and $\mathbf{T}_d^{\mathcal{H}}$ for arbitrary d. At a high level, this section suggests the *modularity and cumulativity* [40] of our tableau systems. Namely, we can switch from logic to logic by simply adding and/or deleting rules in a modular way. We should not that these properties have only been hinted at, and more future work is needed to fully justify this observation.

Also worth mentioning is that we explicitly consider strong completeness, whereas [24] only presents weak completeness.

Section 4 highlights some of the computational advantages of tableau systems for $\mathcal{H}$-valued modal logics. There, we introduced a novel version of prefixed tableaux for the many-valued setting, and defined a prefixed tableau system $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ for the logic $\mathbf{K}^{\mathcal{H}}$. We proved completeness in a constructive manner, and in doing so, constructed a decision procedure (which has been implemented) for $\mathbf{K}^{\mathcal{H}}$ and arrived at a finite frame property. The freedom allowed by the decision procedure (see in particular Remark 4.18) illustrates the *confluence* [40] of our prefixed tableau systems. That is to say, the order in which we select rules to apply does not affect correctness; we can always converge to the same result without backtracking.
The procedure we give is quite powerful, in that it can be modified to construct an $\mathcal{H}$-model for any finite satisfiable set of formulas. Finally, we gave a prefixed tableau system, decision procedure, and finite frame property for the logic $\mathbf{KB}_d^{\mathcal{H}}$ – relying on the fact that our prefixed tableau systems are not destructive.

There are various directions for possible future research. We will briefly mention some now. Firstly, there is a need to closely consider a prefixed tableau system for $\mathbf{K4}_d^{\mathcal{H}}$, as in that case an argument for the termination of the resulting decision procedure is more subtle.

Also immediate is the possibility to study more frame classes, and define tableau systems for the logics of these frame classes. For instance, we may look at ways of combining the rules of $\mathcal{C}\mathbf{K4}_d^{\mathcal{H}}$ and $\mathcal{C}\mathbf{T}_{d'}^{\mathcal{H}}$ to get a tableau system for an $\mathcal{H}$-valued analog of **S4**.
Or, we may wish to introduce new rules to $\mathcal{C}\mathbf{K}^{\mathcal{H}}$ with the aim of obtaining a system for the logic of all d-serial frames.
All of this would contribute towards justifying the claim of modularity and cumulativity. Best would be the presentation of a principled approach to generating unprefixed tableau systems for various classes

of many-valued modal logics. This may relate to a recently established line of structural proof theory research, in which one studies the algorithmic generation of proof systems with good computational properties. Most common in the literature is the generation of generalizations of Gentzen sequent calculi [13, 29]. Attempts to connect the logics and systems studied here to this line of research could be fruitful. Such work may also contribute to the already mentioned need for a formal characterization of the limitations of unprefixed systems.

Another direction for obvious future work is to consider the complexity of the satisfiability and validity problems of the logics in more detail. Further, the confluence of our prefixed systems suggests the potential for heuristics to speed up the convergence of the decision procedure to an answer.

Finally, there is much experimental and applied work that could be done with respect to the implementation. For instance, there is a large community concerned with the testing of automated theorem proving (see for example [52]). There is also the possibility of comparing the propositional fragment of our implementation to other existing many-valued solvers, such as [58]. To make the tool more useful to others, it would also be important to establish if the current implementation could be embedded in some open-source system such as [1] or directly in proof assistants such as Lean. In fact, a recent development in standard modal logic is the implementation in Lean of verified decision procedures based on tableaux [59, 30]. Generalizing such work to the many-valued setting could also pose interesting problems.

References

[1] Pietro Abate and Rajeev Goré. The tableau workbench. *Electronic Notes in Theoretical Computer Science*, 231:55–67, 2009.

[2] Guillermo Badia, Xavier Caicedo, and Carles Noguera. Frame definability in finitely valued modal logics. *Annals of Pure and Applied Logic*, 174(7):103273, 2023.

[3] E. W. Beth. *The Foundations of Mathematics*. North-Holland, Amsterdam, 1959.

[4] Nick Bezhanishvili and Dick de Jongh. Intuitionistic logic. 2006.

[5] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[6] Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*. Elsevier, 2006.

[7] F. Bou, M. Cerami, and F. Esteva. Finite-valued Łukasiewicz Modal Logic is PSPACE-complete. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two*, pages 774–779. AAAI Press, 2011.

[8] Félix Bou, Francesc Esteva, and Lluis Godo. Modal systems based on many-valued logics. In *EUSFLAT Conf.(1)*, pages 177–182, 2007.

[9] Félix Bou, Francesc Esteva, Lluis Godo, and Ricardo Oscar Rodríguez. On the minimum many-valued modal logic over a finite residuated lattice. *Journal of Logic and computation*, 21(5):739–790, 2011.

[10] Cecelia Britz, Willem Conradie, and Wilmarie Morton. Correspondence theory for many-valued modal logic. *arXiv preprint arXiv:2401.07894*, 2024.

[11] Marco Cerami, Francesc Esteva, and Àngel García-Cerdaña. On the relationship between fuzzy description logics and many-valued modal logics. *International Journal of Approximate Reasoning*, 93:372–394, 2018.

[12] Alexander V. Chagrov and Michael Zakharyaschev. Modal logic. In *Oxford logic guides*, 1997.

[13] Agata Ciabattoni, Revantha Ramanayake, and Heinrich Wansing. Hypersequent and display calculi — a unified perspective. *Studia Logica: An International Journal for Symbolic Logic*, 102(6):1245–1294, 2014.

[14] Willem Conradie, Andrew Craig, Alessandra Palmigiano, and Nachoem Wijnberg. Modelling competing theories. In *2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*. Atlantis Press, 2019/08.

[15] Willem Conradie, Dario Della Monica, Emilio Muñoz-Velasco, Guido Sciavicco, and Ionel Eduard Stan. Fuzzy halpern and shoham's interval temporal logics. *Fuzzy Sets and Systems*, 2022.

[16] Willem Conradie, Sabine Frittella, Krishna Manoorkar, Sajad Nazari, Alessandra Palmigiano, Apostolos Tzimoulis, and Nachoem M Wijnberg. Rough concepts. *Information Sciences*, 561:371–413, 2021.

[17] Willem Conradie, Alessandra Palmigiano, Claudette Robinson, Apostolos Tzimoulis, and Nachoem Wijnberg. Modelling socio-political competition. *Fuzzy Sets and Systems*, 407:115–141, 2021.

[18] Marcello D'Agostino, Dov M. Gabbay, Reiner Hähnle, and Joachim Posegga, editors. *Handbook of Tableau Methods*. Springer Netherlands, Dordrecht, 1999.

[19] Christian G. Fermüller and Herbert Langsteiner. Tableaux for finite-valued logics with arbitrary distribution modalities. In *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 156–171. Springer Berlin Heidelberg, 1998.

[20] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Springer Netherlands, Dordrecht, 1983.

[21] Melvin Fitting. Many-valued modal logics. *Fundamenta informaticae*, 15(3-4):235–254, 1991.

[22] Melvin Fitting. Many-Valued Modal Logics II. *Fundamenta Informaticae*, 17(1-2):55–73, January 1992. Publisher: IOS Press.

[23] Melvin Fitting. Many-valued non-monotonic modal logics. In *International Symposium on Logical Foundations of Computer Science*, pages 139–150. Springer, 1992.

[24] Melvin Fitting. Tableaus for many-valued modal logic. *Studia Logica*, 55(1):63–87, February 1995.

[25] Melvin Fitting. *First-order logic and automated theorem proving (2nd ed.)*. Springer-Verlag, Berlin, Heidelberg, 1996.

[26] Melvin Fitting. Modal proof theory. In Patrick Blackburn, Johan Van Benthem, and Frank Wolter, editors, *Studies in Logic and Practical Reasoning*, volume 3 of *Handbook of Modal Logic*, pages 85–138. Elsevier, January 2007.

[27] Melvin Fitting. How True It Is = Who Says It's True. *Studia Logica: An International Journal for Symbolic Logic*, 91(3):335–366, 2009.

[28] S. Frankowski. Definable Classes of Many-Valued Kripke Frames. *Bulletin of the Section of Logic*, 35:27 – 36, 2006.

[29] Sabine Frittella, Giuseppe Greco, Alexander Kurz, Alessandra Palmigiano, and Vlasta Sikimic. Multi-type sequent calculi. *arXiv preprint arXiv:1609.05343*, 2016.

[30] Malvin Gattinger. A verified proof of craig interpolation for basic modal logic via tableaux in lean. *Advances in Modal Logic*, 2022.

[31] Robert Goldblatt. Mathematical modal logic: A view of its evolution. *Journal of Applied Logic*, 1(5):309–392, 2003.

[32] Siegfried Gottwald. *A Treatise on Many-Valued Logics*. Research Studies Press, 2001.

[33] Reiner Hahnle. *Automated Deduction in Multiple-valued Logics*. Oxford University Press, 01 1994.

[34] Petr Hájek. *Metamathematics of Fuzzy Logic*. Springer Dordrecht, 1998.

[35] Petr Hájek and D. Dagmar Harmancová. A many-valued modal logic. In *Proceedings IPMU 96. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1021–1024, 1996.

[36] Petr Hájek, Dagmar Harmancová, Francesc Esteva, Pere Garcia, and Lluís Godo. On modal logics for qualitative possibility in a fuzzy setting. In Ramon Lopez de Mantaras and David Poole, editors, *Uncertainty in Artificial Intelligence*, pages 278–285. Morgan Kaufmann, San Francisco (CA), 1994.

[37] Thomas J. Jech. *The Axiom of Choice*. North-Holland, Amsterdam, Netherlands, 1973.

[38] Dénes König. Über eine schlussweise aus dem endlichen ins unendliche. *Acta Sci. Math.(Szeged)*, 3(2-3):121–130, 1927.

[39] Costas D. Koutras, Christos Nomikos, and Pavlos Peppas. Canonicity and Completeness Results for Many-Valued Modal Logics. *Journal of Applied Non-Classical Logics*, 12(1):7–41, January 2002.

[40] Fabio Massacci. Single Step Tableaux for Modal Logics. *Journal of Automated Reasoning*, 24(3):319–364, 2000.

[41] Charles G Morgan. Local and global operators and many-valued modal logics. *Notre Dame Journal of Formal Logic*, 20(2):401–411, 1979.

[42] Pascal Ostermann. Many-valued modal propositional calculi. *Mathematical Logic Quarterly*, 34(4):343–354, 1988.

[43] Pascal Ostermann. Many-valued modal logics: Uses and predicate calculus. *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik*, 36(4):367–376, 1990.

[44] Graham Priest. Many-valued modal logics: A simple approach. *The Review of Symbolic Logic*, 1(2):190–203, 2008.

[45] Helena Rasiowa and Roman Sikorski. *The Mathematics of Metamathematics*. Polish Scientific Publ., 1968.

[46] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12(4):403–423, November 1983.

[47] Nicholas Rescher. *Many-Valued Logic*, pages 54–125. Springer Netherlands, Dordrecht, 1968.

[48] J. Sakalauskaitė. Tableaus with invertible rules for many-valued modal propositional logics. *Lithuanian Mathematical Journal*, 42:191–203, 2002.

[49] Krister Segerberg. Some modal logics based on a three-valued logic. *Theoria*, 33(1):53–71, 1967.

[50] Raymond M. Smullyan. *First-Order Logic*. Springer, Berlin, Heidelberg, 1968.

[51] Raymond M. Smullyan and Melvin Fitting. *Set Theory and the Continuum Problem*. Dover Publications, Mineola, N.Y, revised edition edition, April 2010.

[52] Geoff Sutcliffe and Christian Suttner. The tptp problem library. *Journal of Automated Reasoning*, 21:177–203, 1998.

[53] Mitio Takano. Subformula property in many-valued modal logics. *The Journal of Symbolic Logic*, 59(4):1263–1273.

[54] Alfred Tarski. Fundamentale Begriffe der Methodologie der deduktiven Wissenschaften. I. *Monatshefte für Mathematik und Physik*, 37(1):361–404, December 1930.

[55] S. K. Thomason. Possible worlds and many truth values. *Studia Logica: An International Journal for Symbolic Logic*, 37(2):195–204, 1978.

[56] Amanda Vidal. On transitive modal many-valued logics. *Fuzzy Sets Syst.*, 407:97–114, 2021.

[57] Amanda Vidal. Undecidability and non-axiomatizability of modal many-valued logics. *The Journal of Symbolic Logic*, 87(4):1576–1605, 2022.

[58] Bou Vidal, Amanda, Félix, and Lluıs Godo. Niblos: A nice bl-logics solver, 2012. Available at `https://www.iiia.csic.es/~amanda/files/TESINA.pdf`.

[59] Minchao Wu and Rajeev Goré. Verified decision procedures for modal logics. In *10th International Conference on Interactive Theorem Proving (ITP 2019)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2019.

[60] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.