Prefixed Tableaux and Decision Procedures for Many-Valued Modal Logics

Guy Axelrod[1,*], Willem Conradie[1,2]

[1]University of the Witwatersrand, South Africa
[2]National Institute for Theoretical and Computational Sciences (NITheCS), South Africa

Abstract

We introduce prefixed tableau systems for many-valued model logics (MVMLs). Semantically, we follow Fitting [1, 2] in allowing both the truth values of propositional variables at states as well as relational links between states in many-valued Kripke frames to take values in an arbitrary, finite Heyting algebra. Fitting [3] introduced tableau systems for these logics which, however, are not amenable to specialization to the MVMLs of certain frame classes, e.g. generalized symmetric frames. We overcome this difficulty through the use of prefixes which keep explicit track of the many-valued accessibility relation constructed on each branch. We prove soundness and completeness of the systems for the MVMLs of the classes of all many-valued frames and all generalized symmetric many-valued frames. We prove that these systems provide decision procedures and discuss and demonstrate their implementations. Further we derive the finite model property for the two logics under consideration.

Keywords

Many-Valued Modal Logic, Prefixed Tableaux, Completeness, Decidability, Finite Model Property

1. Introduction

Many-valued modal logics (MVML) generalize the Kripke semantics of standard modal logic[1] by allowing for many-valued propositional valuations and/or accessibility relations. This is very useful when applying modal logic to reason about problems requiring a logical account of both modality and vagueness. Accordingly, many-valued modal logics have been used to model and reason about problems in a wide range of settings involving different kinds of gradation or vagueness. Fitting [1, 7] suggests that Heyting-valued Kripke models provide natural models of the epistemic stances of committees of experts which elegantly capture the relations of influence or dominance among committee members. In [7], he provides a MVML-based analysis of the 'muddy children puzzle'. Many-valued modal logics are closely related to fuzzy description logics [8], widely applied in the context of the semantic web [9]. In [10], MVML is applied to the task of reasoning about fuzzy temporal relations. Many-valued generalizations of non-distributive modal logics have been employed to model and reason about competition among scientific theories [11] and to capture certain phenomena of socio-political competition [12]. In [13] many-valued modal logics are enlisted into a framework for reasoning about vague-concepts and categorization.

The literature contains numerous different approaches to extending modal logic to a many-valued setting. Some of the earliest proposals are [14, 15, 16, 17, 18]. 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 [1, 2]. The present paper is concerned with the particular approach to MVML established in [2]. There, Fitting introduces $\mathcal{H}$-valued modal logics. More precisely, he defines an interpretation of modal formulas

PAAR'24: 9th Workshop on Practical Aspects of Automated Reasoning, July 2, 2024, Nancy, France
*Corresponding author.
g.axelrod1@gmail.com (G. Axelrod); willem.conradie@wits.ac.za (W. Conradie)
0000-0002-1752-8069 (G. Axelrod); 0000-0001-9906-4132 (W. Conradie)

[1]By 'standard' is meant all those logics studied in standard reference texts in modal logic such as [4, 5, 6]

via generalized Kripke models, in which both propositions and accessibility relations take on values from an arbitrary finite Heyting algebra $\mathcal{H}$. A study of the proof theory of these logics was initiated by Fitting himself when they were first introduced. Specifically, [2] gives a Gentzen sequent calculus for $\mathbf{K}^{\mathcal{H}}$ – the $\mathcal{H}$-valued analogue of the basic modal logic $\mathbf{K}$. Koutras et al. [19] introduce $\mathcal{H}$-frame generalizations of standard Kripke frame properties such as seriality, reflexivity, symmetry and transitivity. 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. They then go on to extend Fitting's sequent calculus for $\mathbf{K}^{\mathcal{H}}$ to sequent calculi for these logics. The sequent calculi in [2] and [19] rely on a cut rule. In [3], Fitting defines a cut-free semantic tableau system for $\mathbf{K}^{\mathcal{H}}$. Extending 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, is relatively straightforward, and is done by the corresponding author in their master's thesis. However, $\mathbf{KB}^{\mathcal{H}}_d$ requires that we introduce prefixes to our tableaux. And, the resulting prefixed systems lend themselves naturally to defining decision procedures.

We now briefly survey some related work. In [20], Priest introduces tableau systems (as well as nice philosophical applications) for certain four and three-valued crisp modal logics. His tableau system is a prefixed one, which, along with the prefixed systems defined in [21], provide the underlying inspiration for the prefixed system presented in this work. More recently, [22] presents what essentially amounts to a prefixed tableau system for a fuzzy version of Halpern and Shoham's Interval Temporal Logic. In [23, 24], a broad basis for the study of MVMLs 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 MVMLs. Vidal has contributed much to this area, and good overviews and references can be found in [25, 26]. Much of this recent work shifts focus from Fitting's finite valued Heyting semantics to more fuzzy, real valued semantics. The works most closely related to what we present here are [27, 28, 29], in that they focus on Fitting's framework. [27] provides a cut-free sequent calculus for $\mathbf{K}^{\mathcal{H}}$, and as such, is essentially the first work to provide a decidability result for this logic. [28] and [29] study tableaux for the crisp versions of the logics we consider here. In particular, [28] provides prefixed tableau systems for such crisp logics with very general modalities. It is not entirely clear how to adapt that work to the non-crisp setting, and the present paper 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 were studied by Hähnle [30, 31].

The paper is structured as follows. In Section 2 we provide the relevant background. Section 3 defines (prefixed) tableaux and presents the system $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$. We go on to prove that $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ is sound wrt the class of all $\mathcal{H}$-frames in Section 4. Section 5 proves the completeness of $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ by way of using the rules to construct a decision procedure for $\mathbf{K}^{\mathcal{H}}$. This also leads us to a finite model property for $\mathbf{K}^{\mathcal{H}}$. Finally, in Section 6, we modify $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ to obtain a prefixed tableau system (and resulting decision procedure and finite model property) for the logic $\mathbf{KB}^{\mathcal{H}}_d$.

2. Background

Analogous to the connection between Boolean algebras and classical propositional logic, Heyting algebras (also called pseudo-Boolean algebras) model the algebraic structure of intuitionistic logic (see [32]). For a detailed exposition of the theory of Heyting algebras and related topics, see [33]. One may approach defining Heyting Algebras either in terms of orderings or purely algebraically. We choose the order theoretic approach.

A partially ordered set $(H, \leq)$ is a ***lattice*** iff every two-element subset $\{a, b\}$ of H has a ***supremum*** (or ***join***), denoted by $a \vee b$, and an ***infimum*** (or ***meet***), denoted by $a \wedge b$. If there exists a least and greatest element of H, then the lattice is said to be ***bounded***. The greatest and least element of any bounded lattice shall be denoted by 0 and 1 respectively. For arbitrary $G \subseteq H$, we define $\bigwedge G := \inf G$ and $\bigvee G := \sup G$. In the case in which G is finite, these objects always exist.

Definition 2.1. A ***Heyting algebra*** $\mathcal{H}$ is a bounded lattice $(H, \leq)$ with the property that for all $a, b \in H$, there exists a $c \in H$ which is the greatest element of $\{c' \in H \mid a \wedge c' \leq b\}$, or equivalently, $d \leq c$ iff $a \wedge d \leq b$

for every $d \in H$. Such a c is unique, and we call it the ***pseudo-complement of*** a ***relative to*** b (and denote it by $a \Rightarrow b$).

Example 2.2. The simplest, non-Boolean Heyting algebra is $\mathcal{H}^3 = (\{0, h, 1\}, \leq)$, where $\leq$ is a total order.

Finite Heyting algebras will serve as the truth value spaces of our logics. The syntax and semantics of the logics we study are parameterized by the specific Heyting algebra we choose to act as the underlying truth value space. So, let us once and for all fix an arbitrary finite Heyting algebra $\mathcal{H} = (H, \leq)$. We continue to use $\wedge, \vee, \Rightarrow$ for the meet, join and relative pseudo-complement. We shall refer to elements of H as ***$\mathcal{H}$-truth values***[2] and include in our language a set of propositional constant $\underline{H} = \{\underline{a} \mid a \in H\}$, one for each element of H. Let us also fix some non-empty countable set Φ of propositional variables. The language for our MVML, which we denote by $\mathcal{L}^{\mathcal{H}}(\Phi)$, consists of finite strings constructed from the alphabet $\underline{H} \cup \Phi \cup \{\wedge, \vee, \supset, \Diamond, \Box, (,)\}$[3]. The set of $\mathcal{H}$-valued modal formulas (or simply 'formulas' from now on), denoted $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$, is generated by the following grammar:

$$\varphi ::= \underline{a} \,| p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \Box\varphi_1 \mid \Diamond\varphi_1$$

where a ranges over $\mathcal{H}$-truth value and p over propositional variables (these are our ***atomic formulas***). For $\varphi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$, the ***modal degree***, denoted $Mdegree(\varphi)$, is the number of occurrences of the symbols $\Diamond$ and $\Box$ in φ. Further, $Sub(\varphi)$ denotes the set of all subformulas of φ.

Formulas will be interpreted in $\mathcal{H}$-valued generalizations of standard Kripke structures. Namely, an ***$\mathcal{H}$-frame*** is a tuple $\mathfrak{F} = (W, R)$, where W is a non-empty set of worlds (or states) and $R : W \times W \to H$ is a function assigning $\mathcal{H}$-truth values to ordered pairs of worlds.

An ***$\mathcal{H}$-model*** 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 $\Phi \cup \underline{H}$. 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, s.t. $V(\mathfrak{s}, \underline{a}) = a$ 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.

We can extend an $\mathcal{H}$-model's valuation to all formulas in $Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ via a recursive definition.

Definition 2.3. 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 $\mathfrak{M}$ 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 $\mathfrak{M} \nVdash \varphi$.

It should be noted that if $\mathcal{H}$ is the Boolean algebra **2** consisting of two elements, then the MVML we have introduced reduces to the standard two-valued modal logic. In this standard case, it is clear that some of our

[2]We will often drop the $\mathcal{H}$ and just speak of ***truth values***.

[3]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. This, in turn, allows us to differentiate between formulas such as $(\underline{a \wedge t_n} \supset \varphi)$ vs $(\underline{a} \wedge \underline{t_n} \supset \varphi)$. This becomes important in some tableau rules, for example see rule ($p\mathbf{K}F\Box$).

connectives are redundant. However, in the general case, the connectives we have in our language are not interdefinable. As such, we need to explicitly include them.

We introduce new symbols which have 'negation-like' semantics which will be crucial for our tableaux. Let T and F be two new formal symbols. A ***signed formula*** consists of a formula with either the symbol T or F prepended to it. 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 $\mathfrak{M}, \mathfrak{s} \Vdash \varphi$; or it is $F\varphi$ and $\mathfrak{M}, \mathfrak{s} \nVdash \varphi$.

Definition 2.4 (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 $\mathfrak{M} \Vdash \varphi$. 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*** $\mathcal{F}$.

We denote the logic of all $\mathcal{H}$-frames by $\mathbf{K}^{\mathcal{H}}$. In the context of standard modal logic, various other classes of frames have been characterized in terms of conditions on the two-valued accessibility relation and extensively studied. Classes of $\mathcal{H}$-frames which are characterized by many-valued generalizations of some of these conditions are defined in [19]. These conditions on the many-valued accessibility relation are parameterized by an arbitrary $\mathcal{H}$-truth value d. In the case of 'many-valued symmetry', we say that an $\mathcal{H}$-frame (W, R) is d-***symmetric*** iff $d \wedge R(\mathfrak{s}, \mathfrak{v}) = d \wedge R(\mathfrak{v}, \mathfrak{s})$ for every $\mathfrak{s}, \mathfrak{v} \in W$. Letting $\mathsf{Symm}^{\mathcal{H}}_d$ denote the class of all d-symmetric $\mathcal{H}$-frames, we use $\mathbf{KB}^{\mathcal{H}}_d$ to denote[4] $\Lambda_{\mathsf{Symm}^{\mathcal{H}}_d}$.

3. Prefixed Tableaux

Tableau systems were first introduced by Beth [35] and popularized by Smullyan [36]. They have since been widely adapted to be used for various non-classical logics [31]. Fitting gives a detailed account of their use for standard modal logics in [21], and this particular text motivated much of the work in this paper.

Before precisely defining prefixed tableaux, we need to define the relevant object language, i.e. the set of strings that can occur in the derivations in our system. First and foremost, we will make use of signed bounding implications, which, as the name suggests, provide a syntactic means by which we can 'bound' the value of a formula. More precisely, a formula is a ***bounding implication*** iff it is of the form $\underline{a} \supset \psi$ or $\psi \supset \underline{a}$ for some $\underline{a} \in \underline{H}$ and $\psi \in Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$.

For a formula φ, it will also be useful to talk about the ***bounded subformulas*** of φ, which are the bounding implications of the form $\underline{a} \supset \psi$ or $\psi \supset \underline{a}$, where $\underline{a} \in \underline{H}$ and $\psi \in Sub(\varphi)$[5].

A ***signed bounding implication*** is simply a signed formula in which the formula is a bounding implication. We denote the set of all signed bounding implications by SBI, and say that $\beta \in SBI$ ***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 $F(\varphi \supset \underline{a})$. We shall use $\perp$ as an abbreviation for $F(\underline{0} \supset \underline{1})$.

Our system expands on the tableau system defined by Fitting in [3]. There, the object language is SBI. We shall be concerned with an object language in which elements of SBI are augmented with ***prefixes***. Fixing some countably infinite set of symbols Σ, a prefix is a tuple (w, σ), where $w \in \Sigma$ and $\sigma \subseteq \Sigma \times \Sigma \times \underline{H}$. 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 $pSBI$, and this will play the role of object language for what we call ***prefixed*** tableaux.

The system in [3] is in the tradition of Smullyan [36], and Fitting presents his (unprefixed) tableaux as trees where each node is labelled by a single element of SBI. However, although not explicitly stated by Fitting, the destructive nature of his modal rules requires that, technically, tableaux are more abstract objects than trees.

[4]The names of the logics are in keeping with convention, as the definitions collapse to the standard case when $d = 1$ and $\mathcal{H} = \mathbf{2}$. For instance, $\mathbf{KB}^{\mathbf{2}}_1$ is the same as the standard modal logic **KB** of symmetric 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 are still defined by them. [34] gives a good account of why this is so.

[5]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.

Specifically, a tableau in [3] is a collection in $\mathcal{P}(\mathcal{P}(SBI))$ (i.e., a set of sets of signed bounding implications). We will use this abstract approach to define prefixed tableaux too. That is to say, the set of prefixed tableaux for some formula will be defined recursively as a subset of $\mathcal{P}(\mathcal{P}(pSBI))$ 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.

$$(\rho)\ \frac{\mathcal{N}}{\mathcal{D}_1 \quad \Big| \quad \dots \quad \Big| \quad \mathcal{D}_n}$$

side condition

The numerator, denominators and side condition of a tableau rule are expressions of the metalanguage. They describe subsets of $pSBI$ 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, s.t. the *side condition* is satisfied. As mentioned, the purpose of a tableau rule $\rho = (\mathcal{N}, (\mathcal{D}_1, \dots, \mathcal{D}_n), \textit{side condition})$ is to describe a family of operations that can be applied to elements of $\mathcal{P}(\mathcal{P}(pSBI))$. To be more precise, let $f : \mathcal{P}(\mathcal{P}(pSBI)) \to \mathcal{P}(\mathcal{P}(pSBI))$. We say f is ***described*** by ρ iff for all $T \in \mathcal{P}(\mathcal{P}(pSBI))$, if $T \neq f(T)$ then for some $S \in T$, S is an instantiation of $\mathcal{N}$, $f(T)$ contains $S_1, \dots S_n$ which are corresponding instantiations of $\mathcal{D}_1, \dots, \mathcal{D}_n$ respectively, and $T \setminus \{S\} = f(T) \setminus \{S_1, \dots, S_n\}$.[6]
In most cases we will not make explicit reference to an operation described by a rule. If $T^* = f(T)$ for some $T \in \mathcal{P}(\mathcal{P}(pSBI))$ and f described by ρ, we shall say that T^* was derived from T through an ***application*** of ρ. 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^*.

Now, we call any finite collection of tableau rules, $\mathcal{C}$, a tableau system.

Definition 3.1. Let X be a finite subset of $pSBI$. The set of $\mathcal{C}$-***tableaux for*** X is a subset of $\mathcal{P}(\mathcal{P}(pSBI))$ and is defined recursively as follows.

- $\{X\}$ is a $\mathcal{C}$-tableau for X
- Suppose T is a $\mathcal{C}$-tableau for X. If $T^* \in \mathcal{P}(\mathcal{P}(pSBI))$ 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 $T \in \mathcal{P}(\mathcal{P}(pSBI))$ s.t. T is a $\mathcal{C}$-tableau for some finite $X \subseteq pSBI$. We call the sets in a $\mathcal{C}$-tableau its ***branches***[7].

Given some set $S \in \mathcal{P}(pSBI)$, we shall say that S is ***closed*** iff $(w, \varnothing)\bot \in S$ for some $w \in \Sigma$. Otherwise, we say that S is ***open***. A tableau is closed iff all its branches are closed; otherwise it is open. We say that a formula φ is a ***theorem*** of $\mathcal{C}$ iff for some $w \in \Sigma$, there exists a closed $\mathcal{C}$-tableau for $\{(w, \varnothing)F(\underline{1} \supset \varphi)\}$. 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.

The unprefixed tableau systems introduced in [3] view the formulas in a branch as describing the valuation at a specific world of a hypothetical model. The application of certain 'modal' rules corresponds to a change in world with a concomitant loss of much of the information regarding the previous world. This 'destructiveness' makes basing a decision procedure upon this system difficult, and what is more, devising a system that is sound and complete wrt e.g. symmetric frames is impossible. To do the former would require a system of bookkeeping and backtracking. We now introduce "non-destructive" tableau systems with prefixes which take care of this bookkeeping naturally inside the system itself and ensure that 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, and call w a ***world label***. We think of $(w, v, \underline{t}) \in \sigma \subseteq \Sigma \times \Sigma \times \underline{H}$ as saying that the world denoted

[6] Note that the identity operation on $\mathcal{P}(\mathcal{P}(pSBI))$ is described by every rule.
[7] The justification for this terminology will be made explicit in Section 5.1.

by v is accessible from the world denoted by w to degree t. We call $(w, v, \underline{t})$ a ***constraint***. We shall use the following convenient notation. For $\beta \in SBI$, $sf((w,\sigma)\beta) := \beta$; $world((w,\sigma)\beta) := w$; $con((w,\sigma)\beta) := \sigma$; and for a given set $X \subseteq pSBI$, we let $cons(X) := \bigcup_{x \in X} con(x)$ and $worlds(X) := \{world(x) \mid x \in X\}$. With prefixes in hand, we view branches of a tableau as describing an entire hypothetical satisfying model – not just a valuation at a specific world. These intuitions are made precise as follows:

Definition 3.2. Let S be a subset of $pSBI$ 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.

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. That is to say, all the rules extend branches, without deleting anything. While such extending rules are usually presented in the literature without placing the numerator in the denominator, we nonetheless do so here in keeping with our earlier abstract definition of tableau rules. Furthermore, we use σ' as an abbreviation for $cons(\mathcal{N})$ [8].

$$(p\bot_1)\ \frac{X; (w,\sigma)T(\underline{a} \supset \underline{b})}{\mathcal{N}; (w,\varnothing)\bot} \quad \text{Where } a \not\leq b$$

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

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

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

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

Table 1
Closing rules

$$(pF{\geq})\ \frac{X; (w,\sigma)F(\underline{a} \supset \varphi)}{\mathcal{N}; (w,\sigma')T(\varphi \supset \underline{t_1}) \ \big|\ \dots \ \big|\ \mathcal{N}; (w,\sigma')T(\varphi \supset \underline{t_n})}$$

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

$$(pF{\leq})\ \frac{X; (w,\sigma)F(\varphi \supset \underline{a})}{\mathcal{N}; (w,\sigma')T(\underline{u_1} \supset \varphi) \ \big|\ \dots \ \big|\ \mathcal{N}; (w,\sigma')T(\underline{u_k} \supset \varphi)}$$

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

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

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

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

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

Table 2
Reversal rules

[8] In all the 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 5.4) easier to prove.

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

Where $a\neq 0$.

$$(pF\wedge)\ \dfrac{X;(w,\sigma)F(\underline{a}\supset(\varphi\wedge\psi))}{\mathcal{N};(w,\sigma')F(\underline{a}\supset\varphi)\ \mid\ \mathcal{N};(w,\sigma')F(\underline{a}\supset\psi)}$$

Where $a\neq 0$.

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

Where $a\neq 1$.

$$(pF\vee)\ \dfrac{X;(w,\sigma)F((\varphi\vee\psi)\supset\underline{a})}{\mathcal{N};(w,\sigma')F(\varphi\supset\underline{a})\ \mid\ \mathcal{N};(w,\sigma')F(\psi\supset\underline{a})}$$

Where $a\neq 1$.

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

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

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

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

Table 3
Propositional rules

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

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

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

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

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

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. $a\wedge t_i\neq 0$.

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

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. $t_i\Rightarrow a\neq 1$.

Table 4
Modal rules

4. Soundness

Let $\mathcal{F}$ be an arbitrary 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.

To prove $\mathcal{C}$ is sound wrt $\mathcal{F}$, it suffices to show that each rule of $\mathcal{C}$ preserves $\mathcal{F}$-satisfiability.

Lemma 4.1. *ρ preserves $\mathcal{F}$-satisfiability for each $\rho\in p\mathcal{C}\mathbf{K}^{\mathcal{H}}$.*

Proof. We need 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 rule individually. We will do so for $p\mathbf{K}F\Box$; leaving the other cases to the reader.

Case $\rho=p\mathbf{K}F\Box$: Then $\mathcal{N}=X;(w,\sigma)F(\underline{a}\supset\Box\varphi)$ and 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, $a\not\leq\bigwedge\{R(I(w),\mathfrak{s})\Rightarrow V(\mathfrak{s},\varphi)\mid\mathfrak{s}\in W\}$. Thus, for some $\mathfrak{s}\in W$, we have $a\wedge R(I(w),\mathfrak{s})\not\leq V(\mathfrak{s},\varphi)$. Suppose $R(I(w),\mathfrak{s})=t_i\in H$. Let $v\in\Sigma$ be any symbol that is not already in $worlds(\mathcal{N})$. We extend the interpretation I to v. Specifically, consider $I' := I \cup \{(v, \mathfrak{s})\}$, 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 $\mathfrak{M}$, and $\mathcal{D}$ is satisfied under I'. $\square$

5. Completeness

We may now approach proving completeness in much the same way as is done in [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, we do something that was not easily achieved with those systems. We use our prefixed system to 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 will also allow us to prove a finite frame property.

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***. For $S \subseteq pSBI$. We define the relation $R_S := \{((w, v), t) \in \Sigma^2 \times H \mid (w, v, \underline{t}) \in cons(S)\}$.

Definition 5.1. Let $S \subseteq pSBI$. 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 $\sigma' \subseteq \Sigma^2 \times \underline{H}$.
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), $\psi \vee \theta$ or $\Diamond\psi$. 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), $\psi \wedge \theta$, $\psi \supset \theta$ or $\Box\psi$. Then, for some u which is a minimal truth value not below a, $(w, \sigma')T(\underline{u} \supset \varphi) \in S$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

[9] See [37], in which this is done in the context of prefixed systems for standard modal logics.

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: $\psi \vee \theta$ or $\Diamond\psi$. 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: $\psi \wedge \theta$, $\psi \supset \theta$ or $\Box\psi$. 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 $\sigma' \subseteq \Sigma^2 \times \underline{H}$.

We will mainly be concerned with this definition in the context 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 5.1).

Lemma 5.2. *If $S \subseteq pSBI$ is downward saturated, then S is satisfiable.*

Proof. Suppose S is downward saturated. 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, $\bigvee\{a \in H \mid (w, \sigma)T(\underline{a} \supset \varphi) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\} \leq V(w, p) \leq \bigwedge\{b \in H \mid (w, \sigma)T(\varphi \supset \underline{b}) \in S \text{ for some } \sigma \subseteq \Sigma^2 \times \underline{H}\}$ [10]. We call $\mathfrak{M}_S$ an $\mathcal{H}$-model ***induced by***[11] S.

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 cases and inductive cases, we need to consider the sub-cases depending on the form of β, which could be $T(\underline{a} \supset \varphi), T(\varphi \supset \underline{a}), F(\underline{a} \supset \varphi)$ or $F(\varphi \supset \underline{a})$. Though there are many, each sub-case is quite routine, and we leave them to the reader.

Once we have established that $P(\varphi)$ holds for all formulas φ, the staisfiablily of S follows easily. For consider the identity map $I : W \to W$. 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 $\beta \in SBI$. 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. □

5.1. Decision Procedure

Essentially, the decision procedure 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. This is possible since, as apposed to the unprefixed systems of [3], none of our rules require us to discard elements in a branch. For us, 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 branch $\mathscr{B}$ in $\mathscr{T}$, we let $U(\mathscr{B}) := \bigcup_n \{U(n)\}$, where n runs over the set of nodes in $\mathscr{B}$.

Let $(\mathscr{T}, U)$ be a labelled tree, and suppose $\{\mathscr{B}^i\}_{i \in I}$ are all of the branches of $\mathscr{T}$. The tableau ***corresponding*** to $(\mathscr{T}, U)$ (denoted $T_{(\mathscr{T},U)}$) is simply the collection $\{U(\mathscr{B}^i)\}_{i \in I}$[12]. We will say that a branch $\mathscr{B}$ of $\mathscr{T}$ is closed

[10] Such a V must exist. For assuming the contrary, 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 $p\bot_5$, contradicting the fact that S is downward saturated.

[11] Note that there may be multiple such models with distinct valuations.

[12] For an arbitrary labelled tree $(\mathscr{T}, U)$, $T_{(\mathscr{T},U)}$ is not necessarily a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau, in the strict sense of Definition 3.1. However, the labeled trees that will crop up in our decision procedure will have the property that $T_{(\mathscr{T},U)}$ is in fact a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{U(r)\}$, where r is the root node of $\mathscr{T}$ (see Lemma 5.5).

iff $U(\mathscr{B})$ is closed. Otherwise, we say that $\mathscr{B}$ is open. We will say that $(\mathscr{T}, U)$ is closed iff all the branches of $\mathscr{T}$ are closed; otherwise, we say it is open. Let us also 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 $(\mathscr{T}, U)$' as a shorthand for actually saying that we extend $(\mathscr{T}, U)$ s.t. the corresponding tableau is derivable via an application of ρ to $T_{(\mathscr{T},U)}$. Suppose $T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau. Let $\rho \in p\mathcal{C}\mathbf{K}^{\mathcal{H}}$, and suppose $T^*_{(\mathscr{T},U)}$ is some $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau derived from $T_{(\mathscr{T},U)}$ via an application of ρ. Then, any labeled tree $(\mathscr{T}^*, U^*)$ extending $(\mathscr{T}, U)$ for which $T_{(\mathscr{T}^*,U^*)} = T^*_{(\mathscr{T},U)}$ can be said to have been derived via an ***application*** of ρ to $(\mathscr{T}, U)$. Further, If $\mathscr{B}$ is a branch of $\mathscr{T}$ but not of $\mathscr{T}^*$, we say that ρ was applied to branch $\mathscr{B}$.

procedure $\textsc{isValid}(\varphi)$ returns **true** or **false**

Require: formula φ
$\alpha := F(\underline{1} \supset \varphi)$
$(\mathscr{T}, U) := \textsc{constructTableau}(\alpha)$
if $(\mathscr{T}, U)$ is closed then **return true**, **else return false**

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

Require: $\alpha \in SBI$
Initialize a labeled tree $(\mathscr{T}, U)$ with root node r and $U(r) := (w_0, \varnothing)\alpha$ ▷ 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
$i := 0$
$(\mathscr{T}_i, U_i) := (\mathscr{T}, U)$
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 $(w, \varnothing)\bot$.
      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 $t \in max(\{c \in H \mid a \nleq c\})$ **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**…
    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 $t \in max(\{c \in H \mid a \nleq c\})$ **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**…
    else if β is of the form $T(\underline{a} \supset (\varphi \wedge \psi))$ for some truth value $a \neq 0$ **then**…
    else if β is of the form $F(\underline{a} \supset (\varphi \wedge \psi))$ for some truth value $a \neq 0$ **then**…
    else if β is of the form $T((\varphi \vee \psi) \supset \underline{a})$ for some truth value $a \neq 1$ **then**…
    else if β is of the form $F((\varphi \vee \psi) \supset \underline{a})$ for some truth value $a \neq 1$ **then**…

else if β is of the form $F(\underline{a} \supset (\varphi \supset \psi))$ **then**
for each $t \in \{c \in H \mid c \leq a$ 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 $t \in \{c \in H \mid c \leq a$ 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 $t \in H$ s.t. $(w, v, \underline{t}) \in \sigma'$ **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**…
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 $t \in H$ s.t. $a \wedge t \neq 0$ **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**…
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 $t \in H$ s.t. $(w, v, \underline{t}) \in \sigma'$ **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**…
end if
end for
end for

We omit some of the steps[13], but the steps we do give illustrate the general theme: we are greedily applying rules to a branch of the labeled tree $(\mathscr{T}, U)$ with the aim of making a specific condition of Definition 5.1 hold for the set of labels in that branch. $(\mathscr{T}_i, U_i)$ denotes the labeled tree immediately 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 [14].

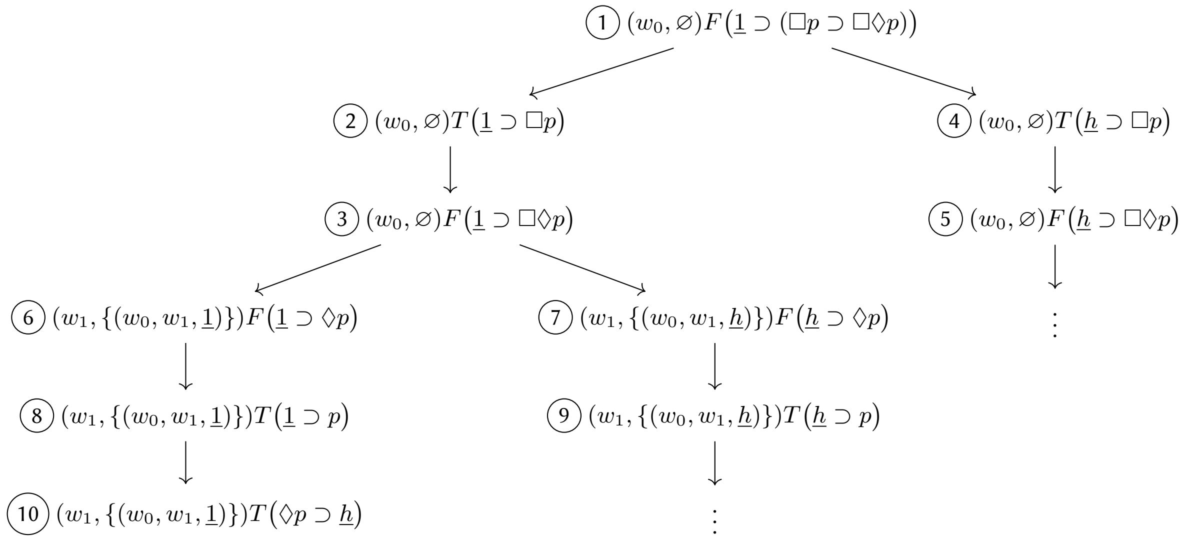

Figure 5.1: Labeled tree constructed during execution of CONSTRUCTTABLEAU$(F(\underline{1} \supset (\Box p \supset \Box\Diamond p)))$

Example 5.3. Assume $\mathcal{H} = \mathcal{H}^3$, and $\varphi = \Box p \supset \Box\Diamond p$. Then ISVALID$(\varphi)$ returns **false**. Let us see why by going through the steps of the procedure. Line 2 of ISVALID(φ) invokes CONSTRUCTTABLEAU$(F(\underline{1} \supset \varphi))$. By stepping through the iterations of the while loop, we can see how we construct the labeled tree shown in Figure 5.1 above. $(\mathscr{T}, U)$ 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 $(w_0, \varnothing)F(\underline{1} \supset \varphi)$ and 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 $(\mathscr{T}, U)$.

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 $w = w_0, \sigma = \varnothing, \beta = F(\underline{1} \supset \varphi)$, 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 32 is met. So, the steps in lines 33 to 36 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 $\mathscr{T}_0$ [15] 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 65 sets $(\mathscr{T}_1, U_1)$ to the current state of $(\mathscr{T}, U)$.

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. Assuming we pick the unfinished node 2 in line 6, the rest of the iteration amounts to performing an identity application of $p\mathbf{K}T\Box$ to $\mathscr{B}_1^3$.

[13]Omission is indicated by ellipses.
[14]As, for instance, will often be the case whenever we reach line 2 in REACTIVATE.
[15]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.

We return to the start of the while loop. $(\mathscr{T}_2, U_2)$ 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 54 is met. So, the steps in lines 55 to 60 are performed. Lines 55 to 59 amount to an application of $p\mathbf{K}F\Box$, which adds nodes 6 and 7 to $\mathscr{T}$. In line 60, 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. In the current context, it leads us to adding nodes 8 and 9 to $\mathscr{T}$, reflecting (non-identity) applications of $p\mathbf{K}T\Box$.

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. Assuming we pick node 6 in line 6, the rest of the iteration leads to us adding node 10, reflecting an application of $pF\geq$ to $\mathscr{B}_3^8$.

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. Assuming we pick node 8 in line 6, the rest of the iteration performs no rule applications.

In the 6^{th} iteration of the while loop, assuming we pick node 10, no new nodes are added, as we perform an identity application of $p\mathbf{K}T\Diamond$ (since there are no $(w, v, \underline{t}) \in \sigma'$ for $w = w_1$).

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 loop 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 5.6). So, as in the proof of Lemma 5.2, $U(\mathscr{B})$ induces an $\mathcal{H}^3$-model $\mathfrak{M}_{U(\mathscr{B})}$, which can be represented as a labelled, weighted, directed graph as follows:

$$w_0 \xrightarrow{\;\;1\;\;} w_1 \quad V(w_1, p) = 1$$

Where we exclude 0-weighted edges and the absence of a label for w_0 indicates that the valuation of propositions at that world can take on any value. As the reader can confirm, evaluating φ at w_0 gives 0. And so, this model is indeed 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 $\{(w_0, \varnothing)F(\underline{1} \supset \varphi)\}$. As we shall see, this observation is a special case of Lemma 5.5.

The following is apparent in general. 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, then for every node n' added to $\mathscr{B}$ in iteration $j \leq i$, we have $con(U(n')) \subseteq con(U(n))$.

We use König's Lemma [38] (see [36, p. 32]) to prove termination. Recall that König's Lemma states: *Every infinite, finitely generated tree must contain at least one infinite branch.* Where a tree is said to be ***finitely generated*** iff every node has a finite number of children.

Proposition 5.4. *For all formulas φ, isValid(φ) terminates.*

Proof. 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 $\mathscr{T}$. 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, $\mathscr{T}$ 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 $sf(x)$ is a signed bounded subformula of φ for all $x \in U(\mathscr{B})$.

For each $k \in \mathbb{N}$, let us define $A_k := \{x \in U(\mathscr{B}) \mid con(x) \text{ has at most } k \text{ elements}\}$, and $B_k := \{x \in U(\mathscr{B}) \mid con(x) \text{ has exactly } k \text{ elements}\}$. Firstly, we can argue by induction that $|worlds(A_k)| \leq k+1$ for every $k \in \mathbb{N}$.

Now consider an arbitrary $k \in \mathbb{N}$. We show that B_k is finite. Since $worlds(A_k)$ is finite, $worlds(B_k)$ (which is a subset of $worlds(A_k)$) is finite. Let $x, x' \in B_k'$. So, $|con(x)| = |con(x')|$, 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 $con(x') \subseteq con(x)$ and so 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. There are only finitely many such x. Thus, B_k must be finite.

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[16]. For the sake of simplifying this proof, let us assume that we pick an unfinished node with a label that has the maximum $Mdegree$ 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. $\square$

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 $\mathscr{T}_i$ s.t. $\mathscr{B}_i$ is a subpath of $\mathscr{B}_j$ starting at the root. And, $U(\mathscr{B}_i) \subseteq U(\mathscr{B}_j)$.

Lemma 5.5. *Let $\alpha \in SBI$. For the labeled tree $(\mathscr{T}, U)$ returned by* constructTableau(α), *$T_{(\mathscr{T},U)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for $\{(w_0, \varnothing)\alpha\}$.*

Proof. We can prove that the following is a loop invariant for the while loop performed by construct-Tableau(α): $T_{(\mathscr{T}_i,U_i)}$ is a $p\mathcal{C}\mathbf{K}^{\mathcal{H}}$-tableau for$\{(w_0, \varnothing)\alpha\}$.

Then, 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 the loop invariant. $\square$

Proposition 5.6. *For all formulas φ,* isValid(φ) *returns **true** iff φ is valid.*

Proof. The forward implication follows from Lemma 5.5 and soundness.

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 it 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 5.1 holds for $U(\mathscr{B})$. This should not be surprising, since the applications of rules in constructTableau are essentially guided by the aim of ensuring that this claim holds. If $U(\mathscr{B})$ is in fact downward saturated, then, by Lemma 5.2, $U(\mathscr{B})$ is satisfiable. But $(w_0, \varnothing)F(\underline{1} \supset \varphi) \in U(\mathscr{B})$, and hence φ cannot be valid. For illustrative purposes, let us confirm here that Condition (9) holds:

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 $1 \leq j \leq k$ s.t. $(w, v, \underline{t}) \in cons(U(\mathscr{B}_j))$. We have two cases. If $j < i$, then $(w, v, \underline{t}) \in cons(U(\mathscr{B}_j)) \subseteq cons(U(\mathscr{B}_{i-1}))$, and the steps in lines 48 to 52 performed for $\mathscr{B}_{i-1}$ ensure that $(v, \sigma')T(\underline{a \wedge t} \supset \varphi) \in U(\mathscr{B})$ for some $\sigma' \subseteq \Sigma^2 \times \underline{H}$. If $j \geq i$, 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

[16]Not all such methods are equally efficient though, since the unfinished node we pick at a given stage can dramatically influence the subsequent size of the constructed tableau.

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 $U(\mathscr{B})$. $\square$

Corollary 5.7. *$p\mathcal{C}\mathbf{K}^{\mathcal{H}}$ is (weakly) complete wrt 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 5.5, 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 5.6, we can conclude that φ is not valid. $\square$

Propositions 5.4 and 5.6 amount to saying that isValid is a decision procedure for the logic $\mathbf{K}^{\mathcal{H}}$. A concrete implementation has been written in python and is provided as a package on PyPi. The source, along with documentation, is available on GitHub (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. And, $\Lambda \subseteq Frm(\mathcal{L}^{\mathcal{H}}(\Phi))$ is said to have the ***finite frame property*** iff $\Lambda = \Lambda_{\mathcal{F}}$ for some class of frames $\mathcal{F}$ of finite character.

Corollary 5.8. *$\mathbf{K}^{\mathcal{H}}$ has the finite frame property, and hence the finite model 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 5.6, 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. $\mathfrak{M}_{U(\mathscr{B})}$ is based on an $\mathcal{H}$-frame (W, R) where $W = worlds(U(\mathscr{B}))$. 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 $\varphi \notin \Lambda_{\mathcal{F}}$. $\square$

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

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 $\mathbf{KB}_d^{\mathcal{H}}$ for all $d \in H$. Let us fix an arbitrary $d \in H$. We proceed to argue that the tableau system

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

is sound and complete wrt $\mathsf{Symm}_d^{\mathcal{H}}$. $p\mathbf{KB}F\Box_d$ and $p\mathbf{KB}F\Diamond_d$ are defined as follows:

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

$$\frac{X; (w, \sigma)F(\underline{a} \supset \Box\varphi)}{\begin{array}{l|c|l|c|l|c|l} \begin{array}{l}\mathcal{N}; (v, \sigma' \cup \\ \{(w, v, \underline{t_1}), (v, w, \underline{t_1^1})\}) \\ F(\underline{a \wedge t_1} \supset \varphi)\end{array} & \ldots & \begin{array}{l}\mathcal{N}; (v, \sigma' \cup \\ \{(w, v, \underline{t_1}), (v, w, \underline{t_1^{k_1}})\}) \\ F(\underline{a \wedge t_1} \supset \varphi)\end{array} & \ldots & \begin{array}{l}\mathcal{N}; (v, \sigma' \cup \\ \{(w, v, \underline{t_n}), (v, w, \underline{t_n^1})\}) \\ F(\underline{a \wedge t_n} \supset \varphi)\end{array} & \ldots & \begin{array}{l}\mathcal{N}; (v, \sigma' \cup \\ \{(w, v, \underline{t_n}), (v, w, \underline{t_n^{k_n}})\}) \\ F(\underline{a \wedge t_n} \supset \varphi)\end{array} \end{array}}$$

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 $i \in \{1, \ldots, n\}$, $\{t_i^1, \ldots, t_i^{k_i}\} = \{t \in H \mid d \wedge t_i = d \wedge t\}$.

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

$$\frac{X;(w,\sigma)F(\Diamond\varphi\supset\underline{a})}{\begin{array}{l|c|l|c|l|c|l} \begin{array}{l}\mathcal{N};(v,\sigma'\cup\\ \{(w,v,\underline{t_1}),(v,w,\underline{t_1^1})\})\\ F(\varphi\supset\underline{t_1\Rightarrow a})\end{array} & \dots & \begin{array}{l}\mathcal{N};(v,\sigma'\cup\\ \{(w,v,\underline{t_1}),(v,w,\underline{t_1^{k_1}})\})\\ F(\varphi\supset\underline{t_1\Rightarrow a})\end{array} & \dots & \begin{array}{l}\mathcal{N};(v,\sigma'\cup\\ \{(w,v,\underline{t_n}),(v,w,\underline{t_n^1})\})\\ F(\varphi\supset\underline{t_n\Rightarrow a})\end{array} & \dots & \begin{array}{l}\mathcal{N};(v,\sigma'\cup\\ \{(w,v,\underline{t_n}),(v,w,\underline{t_n^{k_n}})\})\\ F(\varphi\supset\underline{t_n\Rightarrow a})\end{array}\end{array}}$$

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

Proposition 6.1. *$p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ is sound wrt Symm$_d^{\mathcal{H}}$.*

Proof. It suffices to show that each rule in $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ preserves 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 Symm$_d^{\mathcal{H}}$-satisfiable. That is, there exists an $\mathcal{H}$-model $\mathfrak{M}=((W,R),V)$ based on a frame from 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 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 4.1, with $\mathcal{F}=$ Symm$_d^{\mathcal{H}}$. So consider $\rho=p\mathbf{KB}F\Box_d$. Then $\mathcal{N}=X;(w,\sigma)F(\underline{a}\supset\Box\varphi)$ and so $F(\underline{a}\supset\Box\varphi)$ is satisfied by $\mathfrak{M}$ at $I(w)$. Thus, for some $\mathfrak{s}\in W$, we have $a\not\leq R(I(w),\mathfrak{s})\Rightarrow V(\mathfrak{s},\varphi)$. Suppose $R(I(w),\mathfrak{s})=t_i\in H$ and $R(\mathfrak{s},I(w))=t\in H$. Clearly $a\wedge t_i\neq 0$. Let $v\in\Sigma$ be any symbol that is not already in $worlds(\mathcal{N})$. We extend the interpretation I to v. Specifically, consider $I':=I\cup\{(v,\mathfrak{s})\}$. 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 $\mathfrak{M}$. The argument for $\rho=p\mathbf{KB}F\Diamond_d$ is similar. $\square$

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 5.1), 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 $t'\in\mathcal{H}$ s.t. $t\wedge d=t'\wedge d$.

If S is downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturated, we may use the same approach as in Lemma 5.2 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 Symm$_d^{\mathcal{H}}$. Hence, S is Symm$_d^{\mathcal{H}}$-satisfiable whenever S is downward $p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$-saturated.

Then, suppose we modify constructTableau by 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. With only slight modifications to the arguments given previously, we can show that the new version of isValid is a decision procedure for $\mathbf{KB}_d^{\mathcal{H}}$. And from this we get the following results.

Proposition 6.2. *$p\mathcal{C}\mathbf{KB}_d^{\mathcal{H}}$ is (weakly) complete wrt Symm$_d^{\mathcal{H}}$.*

Corollary 6.3. *$\mathbf{KB}_d^{\mathcal{H}}$ has the finite frame property [17], and hence the finite model property.*

Acknowledgments

The first author acknowledges financial support from the DSI-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS), South Africa. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the CoE-MaSS.

References

[1] M. Fitting, Many-valued modal logics, Fundamenta informaticae 15 (1991) 235–254.

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

[2] M. Fitting, Many-Valued Modal Logics II, Fundamenta Informaticae 17 (1992) 55–73. URL: https://content.iospress.com/articles/fundamenta-informaticae/fi17-1-2-05. doi:10.3233/FI-1992-171-205, publisher: IOS Press.

[3] M. Fitting, Tableaus for many-valued modal logic, Studia Logica 55 (1995) 63–87. URL: https://doi.org/10.1007/BF01053032. doi:10.1007/BF01053032.

[4] P. Blackburn, M. d. Rijke, Y. Venema, Modal Logic, Cambridge University Press, 2001.

[5] P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier, 2006.

[6] A. V. Chagrov, M. Zakharyaschev, Modal Logic, 1997.

[7] M. Fitting, How True It Is = Who Says It's True, Studia Logica: An International Journal for Symbolic Logic 91 (2009) 335–366. arXiv:40269042.

[8] M. Cerami, F. Esteva, Àngel García-Cerdaña, On the relationship between fuzzy description logics and many-valued modal logics, International Journal of Approximate Reasoning 93 (2018) 372–394. URL: https://www.sciencedirect.com/science/article/pii/S0888613X17306990. doi:https://doi.org/10.1016/j.ijar.2017.11.006.

[9] F. Baader, I. Horrocks, U. Sattler, Description logics as ontology languages for the semantic web, in: Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday, Springer, 2005, pp. 228–248.

[10] W. Conradie, D. Della Monica, E. Muñoz-Velasco, G. Sciavicco, I. E. Stan, Fuzzy Halpern and Shoham's interval temporal logics, Fuzzy Sets and Systems (2022). URL: https://www.sciencedirect.com/science/article/pii/S0165011422002068. doi:https://doi.org/10.1016/j.fss.2022.05.014.

[11] W. Conradie, A. Craig, A. Palmigiano, N. 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. URL: https://doi.org/10.2991/eusflat-19.2019.100. doi:https://doi.org/10.2991/eusflat-19.2019.100.

[12] W. Conradie, A. Palmigiano, C. Robinson, A. Tzimoulis, N. Wijnberg, Modelling socio-political competition, Fuzzy Sets and Systems 407 (2021) 115–141.

[13] W. Conradie, S. Frittella, K. Manoorkar, S. Nazari, A. Palmigiano, A. Tzimoulis, N. M. Wijnberg, Rough concepts, Information Sciences 561 (2021) 371–413.

[14] K. Segerberg, Some modal logics based on a three-valued logic, Theoria 33 (1967) 53–71. doi:10.1111/j.1755-2567.1967.tb00610.x.

[15] S. K. Thomason, Possible worlds and many truth values, Studia Logica: An International Journal for Symbolic Logic 37 (1978) 195–204. URL: http://www.jstor.org/stable/20014897.

[16] C. G. Morgan, Local and global operators and many-valued modal logics., Notre Dame Journal of Formal Logic 20 (1979) 401–411.

[17] P. Ostermann, Many-valued modal propositional calculi, Mathematical Logic Quarterly 34 (1988) 343–354. doi:10.1002/malq.19880340411.

[18] P. Ostermann, Many-valued modal logics: Uses and predicate calculus, Zeitschrift fur mathematische Logik und Grundlagen der Mathematik 36 (1990) 367–376. doi:10.1002/malq.19900360411.

[19] C. D. Koutras, C. Nomikos, P. Peppas, Canonicity and Completeness Results for Many-Valued Modal Logics, Journal of Applied Non-Classical Logics 12 (2002) 7–41. URL: https://www.tandfonline.com/doi/abs/10.3166/jancl.12.7-41. doi:10.3166/jancl.12.7-41.

[20] G. Priest, Many-valued modal logics: A simple approach, The Review of Symbolic Logic 1 (2008) 190–203. doi:10.1017/S1755020308080179.

[21] M. Fitting, Proof Methods for Modal and Intuitionistic Logics, Springer Netherlands, Dordrecht, 1983. URL: http://link.springer.com/10.1007/978-94-017-2794-5. doi:10.1007/978-94-017-2794-5.

[22] W. Conradie, R. Monego, E. Muñoz Velasco, G. Sciavicco, I. E. Stan, A Sound and Complete Tableau System for Fuzzy Halpern and Shoham's Interval Temporal Logic, in: A. Artikis, F. Bruse, L. Hunsberger (Eds.), 30th International Symposium on Temporal Representation and Reasoning (TIME 2023), volume 278 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für

Informatik, Dagstuhl, Germany, 2023, pp. 9:1–9:14. URL: https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.TIME.2023.9. doi:10.4230/LIPIcs.TIME.2023.9.
[23] F. Bou, F. Esteva, L. Godo, Modal systems based on many-valued logics., in: EUSFLAT Conf.(1), 2007, pp. 177–182.
[24] F. Bou, F. Esteva, L. Godo, R. O. Rodríguez, On the minimum many-valued modal logic over a finite residuated lattice, Journal of Logic and computation 21 (2011) 739–790.
[25] A. Vidal, On transitive modal many-valued logics, Fuzzy Sets Syst. 407 (2021) 97–114. URL: https://doi.org/10.1016/j.fss.2020.01.011. doi:10.1016/j.fss.2020.01.011.
[26] A. Vidal, Undecidability and non-axiomatizability of modal many-valued logics, The Journal of Symbolic Logic 87 (2022) 1576–1605. doi:10.1017/jsl.2022.32.
[27] M. Takano, Subformula property in many-valued modal logics, The Journal of Symbolic Logic 59 (1994) 1263–1273.
[28] C. G. Fermüller, H. Langsteiner, Tableaux for finite-valued logics with arbitrary distribution modalities, in: Automated Reasoning with Analytic Tableaux and Related Methods, Springer Berlin Heidelberg, 1998, pp. 156–171.
[29] J. Sakalauskaitė, Tableaus with invertible rules for many-valued modal propositional logics, Lithuanian Mathematical Journal 42 (2002) 191–203.
[30] R. Hahnle, Automated Deduction in Multiple-valued Logics, Oxford University Press, 1994. URL: https://doi.org/10.1093/oso/9780198539896.001.0001. doi:10.1093/oso/9780198539896.001.0001.
[31] M. D'Agostino, D. M. Gabbay, R. Hähnle, J. Posegga (Eds.), Handbook of Tableau Methods, Springer Netherlands, Dordrecht, 1999. URL: http://link.springer.com/10.1007/978-94-017-1754-0. doi:10.1007/978-94-017-1754-0.
[32] N. Bezhanishvili, D. de Jongh, Intuitionistic logic, 2006. URL: https://eprints.illc.uva.nl/id/eprint/200, Lecture Notes.
[33] H. Rasiowa, R. Sikorski, The Mathematics of Metamathematics, Polish Scientific Publ., 1968.
[34] C. Britz, W. Conradie, W. Morton, Correspondence theory for many-valued modal logic, arXiv preprint arXiv:2401.07894 (2024).
[35] E. W. Beth, The Foundations of Mathematics, North-Holland, Amsterdam, 1959.
[36] R. M. Smullyan, First-Order Logic, Springer, Berlin, Heidelberg, 1968. doi:10.1007/978-3-642-86718-7.
[37] M. Fitting, Modal proof theory, in: P. Blackburn, J. Van Benthem, F. Wolter (Eds.), Studies in Logic and Practical Reasoning, volume 3 of *Handbook of Modal Logic*, Elsevier, 2007, pp. 85–138. doi:10.1016/S1570-2464(07)80005-X.
[38] D. König, Über eine schlussweise aus dem endlichen ins unendliche, Acta Sci. Math.(Szeged) 3 (1927) 121–130.