EXPRESSIVITY AND CORRESPONDENCE THEORY OF MANY-VALUED HYBRID LOGIC

A dissertation submitted in fulfilment of the
requirements for the degree of

MASTER OF SCIENCE

of

UNIVERSITY OF JOHANNESBURG

by

TINYIKO SAMUEL BALOYI

SUPERVISOR: PROF W. CONRADIE

CO-SUPERVISOR: DR A. CRAIG

DECEMBER 2019

Abstract

The aim of this dissertation is to identify the construction of models that preserve (in both directions) the truth of hybrid formulas and therefore serve to characterize the expressivity of many-valued hybrid logic based on the framework of Hansen, Bolander and Brauner. We show that generated submodels and bounded morphisms preserve the truth of hybrid formulas in both directions. We also show that bisimilarity implies hybrid equivalence in general, however, the converse is not true in general. The converse is true for a weaker notion of a bisimulation for a special set of models, the image-finite models.

The second significant contribution of this project is to develop the correspondence theory for many-valued hybrid logic. We show that the algorithm ALBA(first developed by Conradie and Palmigiano) can be extended to the many-valued hybrid setting. We call this extension MV-Hybrid ALBA. As a result, we successfully identify a syntactically defined class of hybrid formulas for a many-valued hybrid language, namely inductive formulas, whose members always have a local first-order frame correspondents. This inductive class generalizes the Sahlqvist class. An appropriate duality is obtained between frames in the chosen many-valued hybrid framework and a class of algebras having certain properties in order to extend ALBA to the many-valued hybrid setting.

Acknowledgements

I would like to express my special thanks of gratitude to my supervisor Prof Conradie as well as my co-supervisor Dr Craig. Their constructive criticisms played a huge role from the inception of the dissertation up to the end of it. Thank you for sacrificing your time to read up all the different versions of the dissertation and providing comments and suggestions that improved the dissertation.

Secondly, I would like to thank my family and friends for the support and encouragement they have provided me with throughout the period of this research paper. Not only am I grateful for the support on this research paper, but I am grateful for all the support I have received all my life. Your support has been a pillar of my strength from day one. I thank you very much.

Last, but definitely not least, the financial assistance of the National Research Foundation (NRF) and the UJ bursary towards this research are hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF or UJ.

Contents

Introduction

A property is preserved by a certain relation or an operation, if whenever two structures are linked by the relation or operation, then the second structure has the property if the first one has it. We speak of *invariance* if the property is preserved in both directions. Modal logicians are interested in knowing when two structures are indistinguishable by modal languages. That is, when do these structures *satisfy* exactly the same modal formulas. One of the aims of this dissertation is to identify constructions of models of a many-valued hybrid language that will preserve (in both directions) the truth of hybrid formulas and thus serve to characterize the expressivity of many-valued hybrid logic. The concept of validity abstracts away from the effects of particular valuations allows modal languages to get to grips with frame structure. Viewed as tools for defining frames, every modal formula corresponds to a second-order formula. Sometimes this second-order formula has a first-order equivalent. The second aim of this dissertation is to extend the ALBA algorithm to many-valued hybrid logic and use this algorithm to identify a class of hybrid formulas that always have a first-order correspondent and thus develop the correspondence theory for many-valued hybrid logic.

For 2-valued modal logic, the model constructions of *generated submodels*, *disjoint unions* and *bounded morphisms* preserve the truth of formulas in both directions (see Blackburn, De Rijke and Venema [3]). These model constructions are all special cases of *bisimulation* and it was proved that bisimilarity implies *modal equivalence* (when two models satisfy the same set of modal formulas) but modal equivalence does not imply bisimilarity in general. Modal equivalence implies bisimilarity for a special set of models, called the *image-finite* models (see [3, Theorem 2.24]). Although modal equivalence does not imply bisimilarity, it does imply bisimilarity somewhere else, namely in the *ultrafilter extensions* of the models concerned (see [3, Theorem 2.62]). It is also shown in [3] that *validity* is preserved under the formation of disjoint unions, generated submodels and bounded morphisms, and anti-preserved under ultrafilter extensions. The properties of the four frame constructions (disjoint unions, generated submodels, bounded morphisms, and ultrafilter extensions) we have discussed together constitute necessary and sufficient conditions for a class of frames to be *modally definable*. This is essentially the Goldblatt–Thomason Theorem (see [3, Theorem 3.19]). In [27], Sahlqvist defined a class of modal formulas that always have a first-order frame correspondent and this class is now referred to as the *Sahlqvist class*. For formulas of the Sahlqvist class, the corresponding first-order formulas on frames are effectively computable from the modal formula you begin with. The Sahlqvist class was generalized by Goranko and Vakarelov [20] to a bigger class which is referred to as the *inductive class* of formulas.

Hybrid logic is an extension of modal logic. These languages are syntactically simple, and moreover, they turn out to be a useful tool for describing and reasoning about relational structures. Modal semantics are based on states. We evaluate formulas inside the models, at some state, and use the modalities to scan accessible states. However, modal syntax does not allow us to refer to states themselves. It does not let us reason about states themselves. The inability to reason about states themselves could limit us in many applications. Hybrid logic allows us to access more information about the states we are standing in by the use of *nominals*. Hybrid languages have a long history: nominals were used as far back as the sixties in the work of Prior [26] and Bull [8]. Nominals were reinvented on several occasions (see [2, 1, 4, 5, 6]). Nominals are propositional variables with a special property, namely, nominals are only true at exactly one state in any model. For this reason, nominals are used as names for states. This simple idea gives rise to richer logics. As in modal logic, truth of hybrid formulas is invariant under generated submodels and bounded morphisms, but disjoint unions do not preserve the truth of hybrid formulas because of the introduction of *nominals* (see the thesis by Ten Cate [29]). It is this property of being true at exactly one state that causes truth of hybrid formulas to not be preserved under disjoint unions. It was also shown in [29] that bisimilarity implies *hybrid equivalence* but hybrid equivalence only implies bisimilarity under certain conditions, namely, image-finiteness. As in modal logic, hybrid equivalence implies bisimilarity in the ultrafilter extension of models.

Sahlqvist theory in modal logic was generalized to hybrid logic in [29] and later Conradie and Robinson [13] also extended Sahlqvist theory to hybrid logic via an algorithmic approach. The algorithm in question was first developed by Conradie and Palmigiano [10] and makes use of Ackermann's Lemma to eliminate propositional variables. This algorithm is referred to as ALBA where the acronym ALBA stands for "Ackermann's Lemma Based Algorithm".

In classical modal logics, the truth of formulas at a state can only take two values, *true* or *false*. But there is no reason why this has to be the case. A truth-value space could be an infinite set of elements. Each formula can be evaluated at a state to equal an element of the infinite set. This is essentially the many-valued setting. Many-valued modal logics have been considered before in [30, 28, 25, 24, 23]. These papers focused on retaining the general notion of possible state semantics, while allowing formulas to have values in a many-valued space, at each possible state. Fitting [18] extended this basic idea by allowing the accessibility relation between possible states itself to be many-valued.

The model constructions of generated submodels, bounded morphisms and disjoint unions were explored for many-valued modal logic by Eleftheriou and Koutras [16] with a complete Heyting algebra as a truth-value space. In many-valued setting, invariance becomes t-invariance and modal equivalence becomes modal t-equivalence, where t is a nonzero element of the truth-value space. It was shown in [16] that t-invariance results hold for generated submodels, bounded morphisms and disjoint unions. Results about bisimulations for many-valued modal logic were considered by Eleftheriou, Koutras and Nomikos [17]. It was shown that t-bisimilarity implies modal t-equivalence. For the converse, however, with a t-image-finite class of models, a weaker definition of a bisimulation was introduced. A weak bisimulation is a *function* which maps every nonzero element of τ (τ is a complete Heyting algebra) to a set of pairs of states from the two models. The function must satisfy the properties given in Section 3.2 of [17].

The correspondence theory for many-valued modal logic was first explored by Britz [7]. The truth-value space considered in that work is a perfect Heyting algebra. The dissertation focused on generalizing the Sahlqvist theory via an algorithmic approach. In [7], the generalization of ALBA successfully identified a syntactic class of modal formulas (still called the *inductive class* which contained the Sahlqvist class) that always have a local first-order frame correspondent. Furthermore, an interesting fact was discovered: the *restricted Sahlqvist class*, which is subclass of the Sahlqvist class, not only have local first-order frame correspondents but the corresponding local first-order formulas are syntactically identical to the corresponding first-order formulas that modal formulas have in the 2-valued case (see [7, Theorem 4.28]).

Recently, Hansen, Bolander and Brauner introduced many-valued hybrid logic [21]. The basic idea was to combine 2-valued hybrid logic and many-valued modal logic to obtain a many-valued hybrid logic. The accessibility relation between states was also many-valued as in [18]. There is also a function called the *nominal interpreter*. The function maps each nominal to a unique state. This was designed to match the semantics of nominals in the 2-valued hybrid case. In [21], a tableau system that is *sound* and *complete* with respect to the semantics was also defined.

The new work of this dissertation aims to extend the t-invariance results from [16] and [17] to many-valued hybrid logic using a *perfect Heyting algebra* τ (see Definition 2.1.19) as the truth-value space. We will build on the invariance results for 2-valued hybrid logic in [29] and the t-invariance results for many-valued modal logic in [16] to obtain the t-invariance results for generated submodels and bounded morphisms in many-valued hybrid logic. As in 2-valued hybrid logic, we do not have t-invariance results for disjoint unions. For bisimulations, we will build on the invariance results for 2-valued hybrid logic in [29] and t-invariance results for many-valued modal logic in [17] and we will have that t-bisimilarity implies hybrid t-equivalence. For the converse, with a t-image-finite class of models, we also introduced a weaker notion of a bisimulation as in [17] (see Definition 3.4.4) to obtain the results.

The dissertation also seeks to extend the generalized ALBA algorithm to many-valued hybrid logic by combining algorithms presented in [13] for 2-valued hybrid logic and the algorithm in [7] for many-valued modal logic. The ALBA algorithms in [13] and [7] use the fact that modality $\blacklozenge$ is a left adjoint of the modality $\Box$ and modality $\blacksquare$ is a right adjoint of the modality $\Diamond$. The reason for requiring adjoint operators is that the proofs of *soundness* of ALBA rules is based on the adjunction and residuation properties of the connectives as interpreted on *complex algebras*. For our purposes, we need to define the left and right adjoint of our *satisfaction operator* @. The operators $@^\flat$ and $@^\#$ are defined as the right and left adjoints of @, respectively, (see Definition 2.2.10 and Proposition 2.2.12). With these adjoints in our language, we are able to formulate a version of ALBA for our many-valued hybrid logic and refer to that as MV-Hybrid ALBA. We use our MV-Hybrid ALBA to identify an inductive class of formulas that always have a local first-order frame correspondent.

This dissertation is laid out as follows: We will first introduce 2-valued hybrid logic together with 2-valued first-order logic and give a brief description of the correspondence theory between these two logics in Chapter 1. We then review correspondence theory (based on a hybrid version of ALBA) between classical hybrid logic and classical first-order logic. Our description is based on [10]. In Chapter 2, we will introduce many-valued hybrid logic together with its extended many-valued hybrid language. Furthermore, we will also introduce the *algebraic structures* that will be our truth-value space for many-valued hybrid logic and the extended many-valued hybrid language. We will then give a brief introduction of the *duality* between frames and Heyting algebras and define what exactly is the *complex algebra* of a frame. This is needed for the proof of the correctness of MV-Hybrid ALBA. Specifically, we will need a result that says a hybrid formula is true in a frame under a valuation if and only if it is true in the corresponding complex algebra under an assignment (see Proposition 2.3.6).

In Chapter 3, we will explore the expressivity of many-valued hybrid logic. We will show that we have *t*-invariance results for the following model constructions: generated submodels, bounded morphisms and bisimulations in many-valued hybrid logic by building on results from [29] and [16]. For the *t*-invariance under bisimulations, we will build on results from [29] and [17]. In Chapter 4, we will consider the correspondence theory between many-valued hybrid logic and many-valued first-order logic. We will first introduce the many-valued first-order language and the many-valued second-order language. We will then extend the ALBA algorithm to many-valued hybrid logic by building on results from algorithms in [13] and [7]. We will identify a syntactic class of hybrid formulas that always have a local first-order frame correspondent.

Chapter 1

Hybrid Logic

Hybrid logic is an extension of modal logic. Modal logic is an extension of the propositional logic with sentential operators (called *modalities* or *modal operators*). These languages are syntactically simple, and moreover, they turn out to be a useful tool for describing and reasoning about *relational structures*. A relational structure is a tuple consisting of a non-empty set of points and relations between them. These relational structures are widely used in mathematics and computer science.

States are the cornerstone of modal semantics, we evaluate formulas inside the models, at some state, and use the modalities to scan accessible states. However, modal syntax offers no grip on the states themselves. It does not let us reason about states themselves. The inability to reason about state themselves could be a limit to many applications. Hybrid logic allows us to access more information about the states we are standing in by the use *nominals*. Hybrid languages have a long history: nominals were used as far back as the sixties in [26, 8]. Nominals were reinvented at several occasions.

Nominals are the propositional variables with a *special property*, namely, nominals are only true exactly at one state in any model. For this reason, nominals are used as names for states. This simple idea gives rise to richer logics. We now give a brief introduction to hybrid logic based on [3] and [29].

1.1 Syntax and Semantics of Hybrid Logic

Let $\Phi = \{p, q, r, \dots\}$ be a nonempty set of propositional variables and let $\Omega = \{\mathbf{i}, \mathbf{j}, \mathbf{k}, \dots\}$ be a nonempty set disjoint from Φ. The elements are of Ω are called the nominals. We define the *hybrid language* L over the set $\Phi \cup \Omega$ as follows:

$$\phi = p \mid \mathbf{i} \mid \mathbf{0} \mid \neg\phi \mid \phi \vee \psi \mid \Diamond\phi \mid @_{\mathbf{i}}\phi$$

where $p \in \Phi$, $\mathbf{i} \in \Omega$ and for any nominal $\mathbf{i}$, the symbol $@_{\mathbf{i}}$ is called a *satisfaction operator*.

The other logical connectives are defined as one would naturally expect:

- $\phi \wedge \psi = \neg(\neg\phi \vee \neg\psi)$
- $\phi \to \psi = \neg\phi \vee \psi$

We also have a *dual* operator box $\Box$ for our diamond $\Diamond$ operator and it is defined as $\Box\phi = \neg\Diamond\neg\phi$. Note that $\mathbf{0}$ is a falsum in our language and the opposite of that would be the *tautology* and it is defined as $\mathbf{1} = \neg\mathbf{0}$.

Remark 1.1.1. We chose to use the symbols **0** and **1** instead of the natural symbols $\bot$ and $\top$, respectively, because when we introduce the algebraic structures, which will be used as truth-value space for our many-valued hybrid logic the usual choice of top and bottom are **0** and **1**, respectively.

Definition 1.1.2. A *frame* of L is a pair $\mathfrak{F} = (W, R)$ such that $W \neq \emptyset$ and R is a binary relation on W.

Definition 1.1.3. A *model* of L is a pair $\mathfrak{M} = (\mathfrak{F}, V)$ such that $\mathfrak{F}$ is a frame and V is a function assigning to each propositional variable p a subset $V(p)$ and assigning to each nominal $\mathbf{i}$ a singleton $V(\mathbf{i})$. Formally, V is map : $\Phi \cup \Omega \to P(W)$, where $P(W)$ denote the power set of W.

The function V is called a valuation. A valuation assigns to each propositional variable p a subset of W and to each nominal $\mathbf{i}$ a singleton subset of W since a nominal is only true at a *unique state*. That is, a valuation makes propositional variable and nominals true at states. For example, if we have $W = \{a, b, c\}$, $V(p) = \{a, b\}$ specifies that p is true at a and b, but not at c. Also, it is impossible to have a valuation such that $V(\mathbf{i}) = \{a, b\}$. A valuation $V : Form \to P(W)$ can be extended in the following way so we can talk about the truth of every possible formula in our language L.

Definition 1.1.4. Suppose w is a state in a model $\mathfrak{M} = (W, R, V)$. Then we inductively define the notion of a formula ϕ being *true* (or *satisfied*) in $\mathfrak{M}$ at w (notation: $\mathfrak{M}, w \Vdash \phi$) as follows:

1. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$.
2. $\mathfrak{M}, w \Vdash \mathbf{i}$ iff $w \in V(\mathbf{i})$.
3. $\mathfrak{M}, w \Vdash \mathbf{0}$ never the case.
4. $\mathfrak{M}, w \Vdash \neg\phi$ iff $\mathfrak{M}, w \nVdash \phi$.
5. $\mathfrak{M}, w \Vdash \phi \vee \psi$ iff $\mathfrak{M}, w \Vdash \phi$ or $\mathfrak{M}, w \Vdash \psi$.
6. $\mathfrak{M}, w \Vdash \Diamond\phi$ iff there exists $u \in W$ such that Rwu and $\mathfrak{M}, u \Vdash \phi$.
7. $\mathfrak{M}, w \Vdash @_{\mathbf{i}}\phi$ iff there exists $u \in W$ such that $u \in V(\mathbf{i})$ and $\mathfrak{M}, u \Vdash \phi$.

From the semantics of the other connectives, it follows that:

1. $\mathfrak{M}, w \Vdash \mathbf{1}$ is always the case.
2. $\mathfrak{M}, w \Vdash \phi \to \psi$ iff $\mathfrak{M}, w \nVdash \phi$ or $\mathfrak{M}, w \Vdash \psi$.
3. $\mathfrak{M}, w \Vdash \Box\phi$ iff for all $u \in W$ such that Rwu, we have $\mathfrak{M}, u \Vdash \phi$.

Definition 1.1.5. A formula ϕ is *true* in $\mathfrak{M}$ (notation: $\mathfrak{M} \Vdash \phi$) if $\mathfrak{M}, w \Vdash \phi$ for all $w \in W$.

Definition 1.1.6. Let $\mathfrak{F} = (W, R)$ be a frame. A formula ϕ is *valid at a state w in $\mathfrak{F}$* (notation: $\mathfrak{F}, w \Vdash \phi$) if $(\mathfrak{F}, V), w \Vdash \phi$ for all valuations V on $\mathfrak{F}$. A formula ϕ is valid in $\mathfrak{F}$ (notation: $\mathfrak{F} \Vdash \phi$) if it is valid at every state in $\mathfrak{F}$.

As hybrid language is one of the extensions of modal logic, hybrid language can also be extended in a similar fashion. One of the extensions we can have is the hybrid tense logic which adds two *backward looking* modal operators $\blacklozenge$ and $\blacksquare$. The interpretation of hybrid tense logic define events taking place in a series of times so that each state $w \in W$ represents a certain point in time. The intended interpretation of the modal operators of temporal hybrid logic is:

1. $\Diamond\phi$ is interpreted as "ϕ will be true at some future time"
2. $\Box\phi$ is interpreted as "ϕ will be always be true in the future"
3. $\blacklozenge\phi$ is interpreted as "ϕ was true at some past time"
4. $\blacksquare\phi$ is interpreted as "ϕ has always been true in the past"

We will extend our language L by adding these two backward looking modalities for the purpose of hybrid ALBA to be introduced later. For many-valued hybrid ALBA, which will also be introduced later, dual operators for @ operator will also be needed on the extended language as the operator @ can be viewed as both $\Diamond$ and $\Box$ operators in the many-valued setting. The extended language L^+ is defined inductively as

$$\phi = p \mid \mathbf{i} \mid \mathbf{0} \mid \neg\phi \mid \phi \vee \psi \mid \Diamond\phi \mid @_{\mathbf{i}}\phi \mid \blacklozenge\phi \mid \blacksquare\phi$$

We find it convenient to define the following:

Definition 1.1.7. Given formulas $\phi, \phi_1, \ldots, \phi_n$ and $\psi, \psi_1, \ldots, \psi_n$ from L^+, a *quasi-inequality* is an expression of the form $(\phi_1 \leq \psi_1 \& \ldots \& \phi_n \leq \psi_n) \Rightarrow (\phi_n \leq \psi_n)$. Each expression $\phi_i \leq \psi_i$ will be referred to as an *inequality*.

The semantics of L^+ is defined by adding the following clauses to those in Definition 1.1.4.

1. $\mathfrak{M}, w \Vdash \blacklozenge\phi$ iff there exists u such that Ruw and $\mathfrak{M}, u \Vdash \phi$.
2. $\mathfrak{M}, w \Vdash \blacksquare\phi$ iff $\mathfrak{M}, u \Vdash \phi$ for all u such that Ruw.

3. $\mathfrak{M}, w \Vdash \phi \leq \psi$ if and only if $\mathfrak{M}, w \Vdash \phi$ implies that $\mathfrak{M}, w \Vdash \psi$. (Note that the semantics of $\phi \leq \psi$ are the same as the semantics of $\phi \to \psi$)

4. $\mathfrak{M}, w \Vdash (\phi_1 \leq \psi_1 \& \ldots \& \phi_n \leq \psi_n) \Rightarrow (\phi_n \leq \psi_n)$ if and only if $\mathfrak{M}, w \Vdash \phi_i \leq \psi_i$ for all $1 \leq i \leq n$ implies that $\mathfrak{M}, w \Vdash \phi \leq \psi$.

1.2 First-Order Logic

In this section, we will introduce the first-order language that we will denote by L^{FO}. This language is intended to be the correspondence language for our hybrid language L. The content in this section is based on [22]. The description of the language L^{FO} is divided into two steps: symbols and formulas.

1. **Symbols for** L^{FO}: The symbols of $L^{FO}$are divided into two groups:
   (a) Logical symbols:
      i. A set of individual variables VAR, which will be denoted by $x_1, x_2, \ldots$
      ii. Logical connectives $\neg$ and $\vee$
      iii. The quantifier $\exists$
      iv. Equality symbol $=$
   (b) Nonlogical Symbols:
      i. Constant symbols **0** and **1**
      ii. Unary predicate symbols $P_1, P_2, \ldots$
      iii. Constant symbol $c_{\mathbf{i}}$ for each nominal **i**
      iv. A binary relation symbol R

2. **Formulas of** L^{FO}: We first define the *terms* of L^{FO}. The only terms of L^{FO} are the individual variables and constant symbols. With this in our disposal, the formulas of L^{FO} are defined inductively as follows, and this set will be denoted by $FORM$:
   (a) **F1**: If s and t are terms, then $(s = t)$ and $P_n(s)$ are formulas.
   (b) **F2**: If α and β are formulas, so are $\alpha \vee \beta$ and $\neg\alpha$.
   (c) **F3**: If α is a formula and x_n is an individual variable, then $\exists x_n \alpha$ is formula.
   (d) **F4**: Every formula is obtained by a finite number of applications of **F1**, **F2**, **F3**.

We also have the dual operator $\forall$ of $\exists$ such that if α is a formula and x_n is an individual variable, then $\forall x_n \alpha$ is a formula. The connectives $\wedge$ and $\to$ are not part of our language L^{FO}, but as in hybrid language L, we can be define the semantics of these connectives as

1. $\alpha \wedge \beta = \neg(\neg\alpha \vee \neg\beta)$.

2. $\alpha \to \beta = \neg\alpha \vee \beta$.

We now introduce the notion of an interpretation and models in the language L^{FO}. Mathematically, an interpretation has the same structure as frame $\mathfrak{F}$ of hybrid logic L.

Definition 1.2.1. An interpretation I of L^{FO} is a structure consisting of:

1. A non-empty set W^I, called the *domain* of I.

2. For each unary predicate symbol P_n of L^{FO}, a unary predicate P_n^I on W^I.

3. For the binary relation symbol R of L^{FO}, a binary relation R^I on W^I.

The set W^I is also referred to as the *universe* of the interpretation. The variables $x_1, x_2, \ldots$ range over W^I.

We now define the notion of a formula being *true* in an interpretation.

Definition 1.2.2. Let I be an interpretation of L^{FO}. An *assignment in I* is a function v from the set of individual variables into the domain of I, that is,

$$v : VAR \to W^I$$

We can extend v so that it assigns an element W^I to each term of L^{FO}, that is, $v : TERM \to W^I$. We can further extend v so that it assigns a truth value *true* or *false* to each formula A of the language L^{FO}, that is, $v : FORM \to \{0,1\}$. Note that we still denote the extended assignment as v.

Definition 1.2.3. Two assignments v and v' on I are x_n*-variants* (notation : $v' \sim_{x_n} v$) if $v(x_k) = v'(x_k)$ for all $k \neq n$. Thus, x_n-variants v and v' agree everywhere except possibly at x_n, where they may or may not differ.

Definition 1.2.4. We extend the assignment v to formulas of L^{FO} as follows. We denote the extended assignment by $v : FORM \to \{0,1\}$

1. $v(\mathbf{0}) = 0$

2. $v(s = t) = \begin{cases} 1 & \text{if } v(s) = v(t) \\ 0 & \text{otherwise} \end{cases}$

3. $v(R(st)) = \begin{cases} 1 & \text{if } R^I\left(v(s), v(t)\right) \\ 0 & \text{otherwise} \end{cases}$

4. $v\left(P_n\left(s\right)\right) = 1$ iff $P_n^I\left(v(s)\right) = 1$

5. $v\left(\alpha \vee \beta\right) = v\left(\alpha\right) \vee v\left(\beta\right)$

6. $v\left(\exists x_n \alpha\right) = \bigvee\{v'\left(\alpha\right) \mid v' \sim_{x_n} v\}$

The assignment v can also evaluate the following formulas:

1. $v(\mathbf{1}) = 1$
2. $v(\alpha \wedge \beta) = v(\alpha) \wedge v(\beta)$
3. $v(\alpha \rightarrow \beta) = v(\alpha) \rightarrow v(\beta)$
4. $v(\forall x_n \alpha) = \bigwedge \{v'(\alpha) \mid v' \sim_{x_n} v\}$

Remark 1.2.5. Note that we used the same notation v for the extended assignment to all formulas in the logic L^{FO} and we will stick to this notation throughout.

Definition 1.2.6. A formula α of L^{FO} is *true* in an interpretation I (notation : $I \vDash \alpha$) if $v(\alpha) = 1$ under all assignments v in I. We also say that α is *false* in an interpretation I if no assignment satisfies α.

The first-order language L^{FO} is a correspondent language of L. We extend the language L^{FO} with the necessary formulas to acquire the extended first-order language L^{FO+} which will be a suitable correspondent language for L^+.

1.3 The standard translation

The section essentially covers the link between hybrid logic L and first-order logic L^{FO}. The link that connects these logics is called the *standard translation*. The standard translation "translates" a hybrid formula to a corresponding first-order formula. Since a model in L and an interpretation in L^{FO} are mathematically equivalent, we have that a hybrid formula is true in a model if and only if the standard translation of that formula is true in a corresponding first-order interpretation (see Lemma 1.3.4).

Definition 1.3.1. Given a model $\mathfrak{M} = (W, R, V)$ for L, the corresponding first-order interpretation for L^{FO} consists of:

1. A non-empty set W
2. Unary predicate P_n for each propositional variable p_n, where $P_n = V(p_n)$ for each $w \in W$
3. The binary relation R on W

We will also denote the first-order interpretation of a model $\mathfrak{M}$ by $\mathfrak{M}$.

Definition 1.3.2. Let x be a first-order individual variable. The standard translation ST_x taking hybrid formulas of L to formulas L^{FO} is defined by the following:

1. $ST_x(p_n) = P_n(x)$
2. $ST_x(\mathbf{i}) = c_{\mathbf{i}}$

3. $ST_x(\mathbf{0}) = \mathbf{0}$

4. $ST_x(\neg\phi) = \neg ST_x(\phi)$

5. $ST_x(\phi \vee \psi) = ST_x(\phi) \vee ST_x(\psi)$

6. $ST_x(\Diamond\psi) = \exists y\left(Rxy \wedge ST_y(\psi)\right)$

7. $ST_x(@_{\mathbf{i}}\psi) = \exists y\left(y = c_{\mathbf{i}} \wedge ST_y(\psi)\right)$

where y is a fresh variable, that is, a variable that has not been used so far in the translation, $p_n \in \Phi$ and $\mathbf{i} \in \Omega$. We can also define the standard translations of the following formulas:

1. $ST_x(\mathbf{1}) = \mathbf{1}$

2. $ST_x(\phi \wedge \psi) = ST_x(\phi) \wedge ST_x(\psi)$

3. $ST_x(\phi \rightarrow \psi) = ST_x(\phi) \rightarrow ST_x(\psi)$

4. $ST_x(\Box\psi) = \forall y\left(Rxy \rightarrow ST_y(\psi)\right)$

Example 1.3.3. We translate the formula $\phi = @_{\mathbf{i}}p \rightarrow \Box(p \wedge \mathbf{i})$.

$$\begin{aligned} ST_x(@_{\mathbf{i}}p \rightarrow \Box(p \wedge \mathbf{i})) &= ST_x(@_{\mathbf{i}}p) \rightarrow ST_x(\Box(p \wedge \mathbf{i})) \\ &= \exists y\left(y = c_{\mathbf{i}} \wedge ST_y(p)\right) \rightarrow \forall z\left(Rxz \rightarrow ST_z(p \wedge \mathbf{i})\right) \\ &= \exists y\left(y = c_{\mathbf{i}} \wedge P(y)\right) \rightarrow \forall z\left(Rxz \rightarrow (P(z) \wedge c_{\mathbf{i}})\right) \end{aligned}$$

Lemma 1.3.4. Let $\mathfrak{M} = (W, R, V)$ be a model for L, $w \in W$ and ϕ a hybrid formula. Then

1. $\mathfrak{M}, w \Vdash \phi$ iff $\mathfrak{M} \vDash ST_x(\phi)[x := w]$

2. $\mathfrak{M} \Vdash \phi$ iff $\mathfrak{M} \vDash \forall x ST_x(\phi)[x := w]$

The next results say that validity in a frame implies truth in the corresponding second-order structure, and vice versa. The second-order formulas are obtained by quantifying over predicate variables, and not just individual variables.

Lemma 1.3.5. Let $\mathfrak{F} = (W, R)$ be a frame for L, $w \in W$ and ϕ a formula of L. Then:

$$\mathfrak{F}, w \Vdash \phi \text{ iff } \mathfrak{F} \vDash \forall \mathbf{P}\left(ST_x(\phi)\right)[x := w]$$

where $\mathbf{P}$ is the vector of all predicate variables occurring in $ST_x(\phi)$

The last part of this section will assert that the same results hold for the standard translation taking hybrid formulas from L^+ into L^{FO+}.

Definition 1.3.6. Let x be a first-order individual variable. The standard translation ST_x taking hybrid formulas of L^+ to formulas of L^{FO+} deals with additional clauses as follows:

1. $ST_x(\blacklozenge\psi) = \exists y(Ryx \wedge ST_y(\psi))$.

2. $ST_x(\blacksquare\psi) = \forall y(Ryx \rightarrow ST_y(\psi))$.

1.4 Correspondence Theory between Hybrid Logic and First-Order Logic

Frame definability is a second-order notion. Henceforth, the second-order correspondent of any hybrid formula can be straightforwardly computed using the second-order translation. This section aims to explain why many hybrid formulas define first-order conditions on frames.

1.4.1 First-Order Correspondence

Definition 1.4.1. Let ϕ be a hybrid formula in L and $\alpha(x)$ a formula in the corresponding first-order language L^{FO} (x is supposed to be the only free variable of α). Then we say that ϕ and $\alpha(x)$ are *local frame correspondents* of each other if the following holds, for any frame $\mathfrak{F}$ and any state w of $\mathfrak{F}$:

$$\mathfrak{F}, w \Vdash \phi \text{ iff } \mathfrak{F} \vDash \alpha[w]$$

Definition 1.4.2. Let ϕ be a hybrid formula in L and $\alpha(x)$ a formula in the corresponding first-order language L^{FO} (x is supposed to be the only free variable of α). Then we say that ϕ and $\alpha(x)$ are *global frame correspondents* of each other if the following holds, for any frame $\mathfrak{F}$ and for all states of $\mathfrak{F}$:

$$\mathfrak{F} \Vdash \phi \text{ iff } \mathfrak{F} \vDash \alpha$$

Example 1.4.3. We show that the hybrid formula $\phi = @_{\mathbf{i}} p \rightarrow @_{\mathbf{i}} \Diamond p$ globally corresponds to the first-order property reflexivity (Rxx).

Suppose that $\mathfrak{F}$ is a reflexive frame. Let $w \in W$ be a state in $\mathfrak{F}$ and V be any valuation on $\mathfrak{F}$ such that $(\mathfrak{F}, V), w \Vdash @_{\mathbf{i}} p$. Then there exists a state $u \in W$ such that $(\mathfrak{F}, V), u \Vdash p$ and $u \in V(\mathbf{i})$. Since $\mathfrak{F}$ is reflexive, we have that $(\mathfrak{F}, V), u \Vdash \Diamond p$. Hence, by the definition of the $@_{\mathbf{i}}$ operator, we have that $(\mathfrak{F}, V), w \Vdash @_{\mathbf{i}} \Diamond p$.

We prove the other implication by arguing contrapositively. Suppose that we have a frame $\mathfrak{F}$ that is not reflexive. Then we have a point $w \in \mathfrak{F}$ such that $\neg Rww$. Let V be a valuation such that $V(p) = \{w\}$ and $V(\mathbf{i}) = \{w\}$. Hence it follows that $\mathfrak{F}, w \Vdash \Diamond p$. Hence we have that $\mathfrak{F}, w \Vdash @_{\mathbf{i}} \Diamond p$.

1.4.2 Inductive Formulas

The algorithm ALBA was first introduced in [10] for distributive modal logic and later introduced for non-distributive modal logic in [11]. The work was later extended to hybrid logic in [13]. The aim of the algorithm is to eliminate all propositional variables from a given hybrid formula or inequality through application of rules. If this is successful, the standard translation is applied to the set of quasi-inequality produced which gives out the first-order frame correspondent for the given formula or inequality.

This whole section introduces a new syntactically defined class of inequalities in L. This class expands the Sahlqvist class in [3, 27] and *inductive formulas* introduced in [20]. This newly introduced class generalizes the nominalized Sahlqvist-van Benthem formulas in [10]. One of the purpose of this dissertation is to prove that the formulas in this class have first-order local frame correspondents.

Definition 1.4.4. To any formulas $\phi \in L$ we assign two *signed generation trees*, $+\phi$ and $-\phi$, each beginning at the root with the main connective signed, respectively, $+$ and $-$ and then branching out into n-ary connectives. Each leaf is either a propositional variable, a nominal or a truth constant. The nodes are signed as follows:

- the root node of $+\phi$ is signed $+$ and the root node of $-\phi$ is signed $-$;
- if a node is labelled with $\vee, \wedge, \Diamond, \Box$, its children inherit its sign;
- if a node is labelled with $\neg$, its child is assigned the opposite sign;
- if a node is labelled with $\rightarrow$, the right child inherits its sign, while the left child is assigned the opposite sign;
- if a node is labelled with @ (corresponding to a subformula $@_{\mathbf{i}}\alpha$), the right child (corresponding to α) inherits its sign, while the left child (corresponding to $\mathbf{i}$) is assigned the sign $\pm$.

In a generation tree of a formula, certain types of nodes will be regarded as *Skeleton* and others will be regarded as *PIA*. This classification is given in table 1.1:

<table>
<tr><th colspan="5">Skeleton</th><th colspan="4">PIA</th></tr>
<tr><td colspan="5">Δ-adjoints</td><td colspan="4">SRA</td></tr>
<tr><td colspan="2">Primary</td><td colspan="3">Secondary</td><td>+</td><td>□</td><td>∧</td><td>¬</td></tr>
<tr><td colspan="2">+∨</td><td colspan="3">+∧</td><td>−</td><td>◊</td><td>∨</td><td>¬</td></tr>
<tr><td colspan="2">−∧</td><td colspan="3">−∨</td><td colspan="4"></td></tr>
<tr><td colspan="5">SLR</td><td colspan="4">SRR</td></tr>
<tr><td>+</td><td>◊</td><td>¬</td><td>@</td><td></td><td>+</td><td>∨</td><td>@</td><td>→</td></tr>
<tr><td>−</td><td>□</td><td>¬</td><td>@</td><td>→</td><td>−</td><td>∧</td><td>@</td><td></td></tr>
</table>

Table 1.1: Skeleton and PIA nodes

We will further classify them as Δ-adjoints, SLR, SRA and SRR, according to the specification in table 1.1.

Remark 1.4.5. The acronym PIA stands for "positive antecedent implies atom" and is due to Van Benthem [31]. They were introduced because they admit minimal valuations definable in a first-order language. The abbreviations SLR, SRA and SRR stand for *syntactically left residual*, *right adjoint* and *right residual*, respectively. These nodes are classified according to the order-theoretic properties of their interpretations.

We now have a look at how generation trees of hybrid formulas are constructed.

Example 1.4.6. We find the positive generation trees of $@_{\mathbf{i}} p \vee \Diamond(\Diamond q \to p)$ and $@_{\mathbf{i}}(((p \wedge @_{\mathbf{i}} \neg p) \wedge q) \to \Diamond q)$

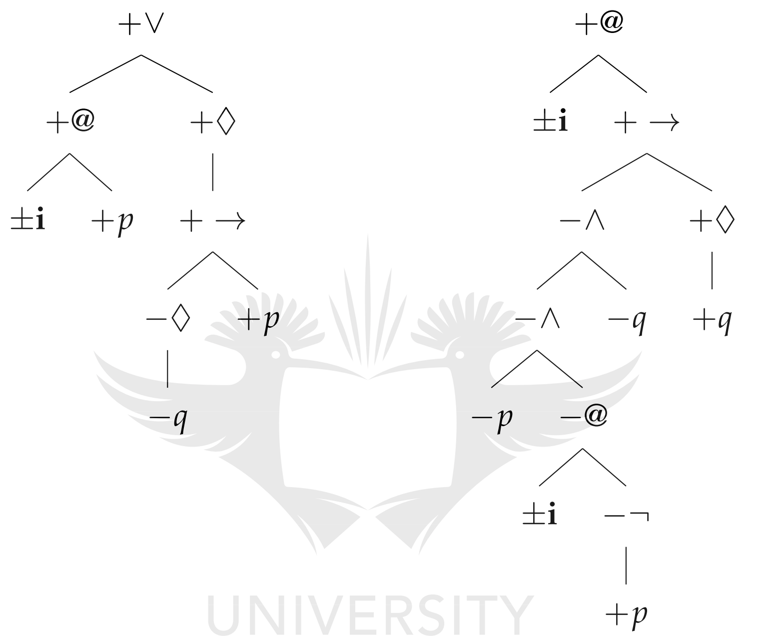


Figure 1.1: left-hand side is a positive generation tree of $@_{\mathbf{i}} p \vee \Diamond(\Diamond q \to p)$ and right-hand side is positive generation tree of $@_{\mathbf{i}}(((p \wedge @_{\mathbf{i}} \neg p) \wedge q) \to \Diamond q)$

We now find the negative generation trees of $@_{\mathbf{i}} p \to \Box \neg q \wedge \mathbf{i}$ and $\mathbf{i} \to \Box(\Diamond \mathbf{i} \to \mathbf{i})$

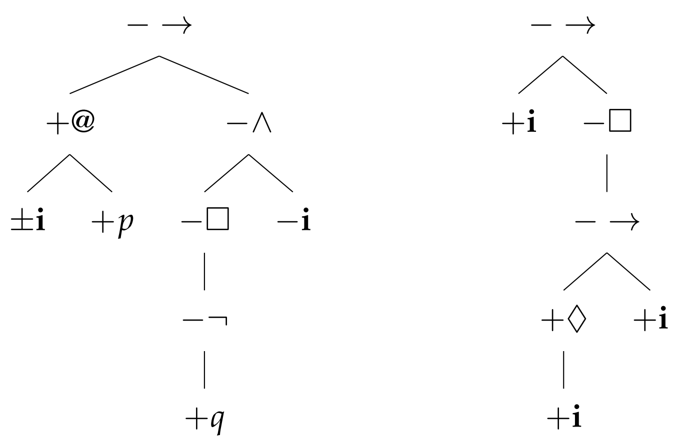


Figure 1.2: left-hand side is a negative generation tree of $@_{\mathbf{i}}p \to \Box\neg q \wedge \mathbf{i}$ and right-hand side is negative generation tree of $\mathbf{i} \to \Box(\Diamond\mathbf{i} \to \mathbf{i})$

Definition 1.4.7. A node in a signed generation tree is said to be *positive* if it is signed "$+$", *negative* if it is signed "$-$" and *bi-polar* if it is signed "$\pm$".

The notion of the positivity and negativity of propositional variables can be equivalently defined as follows:

Remark 1.4.8. A formula ϕ is *positive* (*negative*) in a propositional variable p if every occurrence of p in a leaf of the generation tree $+\phi$ is signed $+(-)$. A formula ϕ is *positive* (*negative*) in a nominal $\mathbf{i}$ if every occurrence of $\mathbf{i}$ in a leaf of the generation tree $+\phi$ is signed $+$ or $\pm$ ($-$ or $\pm$).

Note that the only nodes signed "$\pm$" are those corresponding to the nominal "subscript" argument of the @ operator.

Example 1.4.9. The positive generation tree of the formula $\phi = @_{\mathbf{i}}p \vee \Diamond(\Diamond q \to p)$ in Example 1.4.6 implies that the formula ϕ is positive in the propositional variable p and it is negative in the propositional variable q. Moreover, ϕ can be seen as being positive on the nominal $\mathbf{i}$ and can also be seen as being negative on the nominal $\mathbf{i}$ since the every occurrence of $\mathbf{i}$ in a leaf of a positive generation tree of ϕ is signed $\pm$.

Definition 1.4.10. An *order type* $n \in \mathbb{N}$ is an n-tuple $\varepsilon \in \{1, \delta\}^n$. Given an order type $\varepsilon = (\varepsilon_{p1}, \ldots, \varepsilon_{pn})$, its opposite order type, ε^δ, defined by $\varepsilon^\delta_{pi} = 1$ iff $\varepsilon_{pi} = \delta$ for every $1 \leq i \leq n$.

An order type is used to keep track of designated polarities of each variable in a formula.

Definition 1.4.11. For any formula $\phi(p_1, \ldots, p_n)$, any order type $\varepsilon = (\varepsilon_{p1}, \ldots, \varepsilon_{pn})$, and any $1 \leq i \leq n$, an *ε-critical* node in a signed generation tree of ϕ is a (leaf) node labelled with $+p_i$ if $\varepsilon_i = 1$, or $-p_i$ if $\varepsilon_i = \delta$. An ε-critical *branch* in the tree is a branch terminating in an ε-critical node.

Definition 1.4.12. ([13, Defintion 2.1]) Let $\phi(p_1, \ldots, p_n)$ be a formula in the propositional variables $p_1, \ldots, p_n$, let ε be an order type on $\{1, \ldots, n\}$ and $<_\Gamma$ a *strict partial order* on $p_1, \ldots, p_n$. A branch in a signed generation tree $*\phi, * \in \{+, -\}$, ending in a propositional variable is an (ε, Γ)-*conforming* branch if, apart from the leaf, it is the concatenation of two paths P_1 and P_3, one which may possibly be of length 0, such that P_1 consists only of skeleton modes, and moreover it satisfies the following conditions:

1. For every SRR node in P_1 of the form $\gamma \odot \beta$ or $\beta \odot \gamma$, where β is the side where the branches lies, $\varepsilon^\delta(\gamma) \prec^* \phi$ (that is, γ contains no variables to be solved for)
   In particular:
   (a) if $\gamma \odot \beta$ is $+(\gamma \vee \beta)$ or $+(\beta \to \gamma)$, then $\varepsilon^\delta(+\gamma)$;
   (b) if $\gamma \odot \beta$ is $+(\gamma \to \beta)$ or $-(\gamma \wedge \beta)$, then $\varepsilon^\delta(-\gamma)$ (equivalently, $\varepsilon(+\gamma)$);
   (c) if $\gamma \odot \beta$ is $@_\mathbf{i}\beta$, then condition is met.
2. For every SRR node in P_1 of the form $\gamma \odot \beta$ or $\beta \odot \gamma$, where β is the side where branch lies, $p_j <_\Gamma p_i$ for every p_j occurring in γ, where p_i is the propositional variable labelling the leaf of the branch.

The following definitions are found in [13, Definition 2.2] grouped together as one definition.

Definition 1.4.13. A signed generation tree $*\phi, * \in \{+, -\}$, is said to be $\{\varepsilon, \Gamma\}$-*inductive* if every ε-critical branch in it is (ε, Γ)-conforming.

Definition 1.4.14. A formula ϕ is (ε, Γ)-*inductive* if $-\phi$ is (ε, Γ)-inductive. A formula ϕ is *inductive* if it is (ε, Γ)-inductive for some ε and Γ.

Definition 1.4.15. An inequality $\phi \leq \psi$ is (ε, Γ)-*inductive* if both generation trees $+\phi$ and $-\psi$ are (ε, Γ)-inductive. The inequality $\phi \leq \psi$ is *inductive* if it is (ε, Γ)-inductive for some ε and Γ.

1.4.3 Hybrid-ALBA

In this section we present the hybrid ALBA algorithm which was introduced in [13]. Our aim will be to extend this algorithm to many-valued hybrid ALBA and prove the correctness of the algorithm in the many-valued hybrid setting (this will be done in section 4.2). The algorithm has four phases and each has certain rules used in it. Here are the phases:

- Phase 1: Preprocessing

  The aim of this phase is to equivalently break up an inequality $\phi \leq \psi$, given as an input, into smaller inequalities through the application of the rules ($\vee$-Adj) and ($\wedge$-Adj) to be given in the phase 3. To make it easier, consider the positive generation tree of ϕ and the negative generation tree of ψ, and surface positive occurrence of $\vee$ and negative occurrence of $\wedge$ by applying the following standard equivalences:

$$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \qquad \alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
$$\neg (\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta \qquad \neg (\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$
$$\Diamond (\alpha \vee \beta) \equiv \Diamond\alpha \vee \Diamond\beta \qquad \Box (\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$$
$$@_{\mathbf{i}} (\alpha \vee \beta) \equiv @_{\mathbf{i}}\alpha \vee @_{\mathbf{i}}\beta \qquad @_{\mathbf{i}} (\alpha \wedge \beta) \equiv @_{\mathbf{i}}\alpha \wedge @_{\mathbf{i}}\beta$$

Let $\mathit{Preprocess}\,(\phi \leq \psi) = \{\phi_i \leq \psi_i \mid i \in I\}$ be the finite set of inequalities obtained after exhaustive application of the above equivalences.

- Phase 2: First approximation

  Each inequality produced in Phase 1 is turned into a quasi-inequality by applying the following *first approximation rule*. The algorithm now proceeds separately on each of the quasi-inequalities obtained.

  First-approximation. Let $\mathit{Preprocess}\,(\phi \leq \psi) = \{\phi_i \leq \psi_i \mid i \in I\}$ be the set of inequalities obtained in Phase 1. Then the following *first-approximation rule* is applied to each $\phi_i \leq \psi_i$ only once:

$$\frac{\phi_i \leq \psi_i}{\mathbf{m}_0 \leq \phi_i \& \psi_i \leq \neg\mathbf{n}_0 \Rightarrow \mathbf{m}_0 \leq \neg\mathbf{n}_0} \text{ (First-approximation)}$$

  where $\mathbf{m}_0$ and $\mathbf{n}_0$ are *special reserved nominals* which do not occur in any inequality received in input.

  The First-approximation yield the systems of inequalities

  $\{\mathbf{m}_0 \leq \phi_i \& \psi_i \leq \mathbf{n}_0\}$ for each inequality in $\mathit{Preprocess}\,(\phi \leq \psi)$. Each such a system is called an *initial system*.

- Phase 3: Reduction and Elimination
  This Phase focuses on eliminating all the propositional variables from the quasi-inequalities resulting in Phase 2 through the application of the *Ackermann rules* (RH-Ack) and (LH-Ack), or their special case (RH-Ack-0) and (LH-Ack-0). To bring the quasi-inequality into the shape to which one of these rules is applicable, the *approximation*, *residuation* and *adjunction* rules are used. If all propositional variables occurrences have been eliminated, we denote the resulting set of pure quasi-inequalities by $\mathit{pure}\,(\phi, \psi)$. If some propositional variable could not be eliminated, then the algorithm fails.

  Adjunction rules:

$$\frac{\alpha \leq \beta \wedge \gamma}{\alpha \leq \beta \,\&\, \alpha \leq \gamma}\ (\wedge\text{-Adj}) \qquad \frac{\alpha \vee \beta \leq \gamma}{\alpha \leq \gamma \,\&\, \beta \leq \gamma}\ (\vee\text{-Adj})$$

$$\frac{\alpha \leq \Box\beta}{\blacklozenge\alpha \leq \beta}\ (\Box\text{-Adj}) \qquad \frac{\Diamond\alpha \leq \beta}{\alpha \leq \blacksquare\beta}\ (\Diamond\text{-Adj})$$

$$\frac{\alpha \leq \neg\beta}{\beta \leq \neg\alpha}\ (\neg\text{-R-Adj}) \qquad \frac{\neg\alpha \leq \beta}{\neg\beta \leq \alpha}\ (\neg\text{-L-Adj})$$

The rules ($\wedge$-Adj) and ($\vee$-Adj) are justified by the fact that $\wedge$ is a right adjoint and $\vee$ is a left adjoint of the diagonal map $\Delta : \mathbf{L} \to \mathbf{A}\text{x}\mathbf{A}$ given by $\Delta(a) = (a, a)$ and the rules ($\Box$-Adj) and ($\Diamond$-Adj) are justified by the fact that $\Box$ is the right adjoint of $\blacklozenge$ and $\Diamond$ is the left adjoint of $\blacksquare$. The last two rules follow from the fact that $\neg$ is its own adjoint.

Residuation rules:

$$\frac{\alpha \wedge \beta \leq \gamma}{\alpha \leq \beta \to \gamma}\ (\wedge\text{-Res}) \qquad \frac{\alpha \leq \beta \vee \gamma}{\alpha \wedge \neg\beta \leq \gamma}\ (\vee\text{-Res}) \qquad \frac{\alpha \leq \beta \to \gamma}{\alpha \wedge \beta \leq \gamma}\ (\to\text{-Res})$$

$$\frac{\alpha \leq @_{\mathbf{i}}\beta}{\alpha \leq \mathbf{0} \parr \mathbf{i} \leq \beta}\ (@\text{-R-Res}) \qquad \frac{@_{\mathbf{i}}\alpha \leq \beta}{\mathbf{1} \leq \beta \parr \alpha \leq \neg\mathbf{i}}\ (@\text{-L-Res})$$

The residuation rules are based on the residuation properties of the interpretations of the connectives.

Approximation rules:

$$\frac{\Box\alpha \leq \neg\mathbf{i}}{\exists\mathbf{j}\left(\Box\neg\mathbf{j} \leq \neg\mathbf{i} \,\&\, \alpha \leq \neg\mathbf{j}\right)}\ (\Box\text{-approx}) \qquad \frac{\mathbf{i} \leq \Diamond\alpha}{\exists\mathbf{j}\left(\mathbf{i} \leq \Diamond\mathbf{j} \,\&\, \mathbf{j} \leq \alpha\right)}\ (\Diamond\text{-approx})$$

$$\frac{\mathbf{i} \leq @_{\mathbf{j}}\alpha}{\mathbf{j} \leq \alpha}\ (@\text{-R-Approx}) \qquad \frac{@_{\mathbf{j}}\alpha \leq \neg\mathbf{i}}{\alpha \leq \neg\mathbf{j}}\ (@\text{-L-Approx})$$

where the nominal **j** introduced in (□-Approx) and (◇-Approx) is fresh, that is, **j** has not yet occurred thus far in the computation.

The approximation rules follows from the fact that in a complete and *atomic* hybrid algebra each element is the join of atoms below it and the meet of co-atoms above it.

Ackermann rules: Once the application of the adjunction, residuation and approximation rules has turned the system to the desired shape, the Ackermann rules are applied to the whole system to eliminate all the propositional variables. Here the Ackermann rules:

$$\frac{\&_{i=1}^{n} \alpha_i \leq p \,\&\, \&_{j=1}^{m} \beta_j(p) \leq \gamma_j(p)}{\&_{j=1}^{m} \beta_j\left(\bigvee_{i=1}^{n} \alpha_i\right) \leq \gamma_j\left(\bigvee_{i=1}^{n} \alpha_i\right)} \text{(RH-Ack)} \tag{1.1}$$

$$\frac{\&_{i=1}^{n} p \leq \alpha_i \,\&\, \&_{j=1}^{m} \gamma_j(p) \leq \beta_j(p)}{\&_{j=1}^{m} \gamma_j\left(\bigwedge_{i=1}^{n} \alpha_i\right) \leq \beta_j\left(\bigwedge_{i=1}^{n} \alpha_i\right)} \text{(LH-Ack)} \tag{1.2}$$

where

1. the α_i are p-free;
2. the β_j are positive in p; and
3. the γ_j are negative in p.

If $n = 0$, $\bigvee_{i=0}^{n} \alpha_i \equiv \mathbf{0}$ and $\bigwedge_{i=0}^{n} \alpha_i \equiv \mathbf{1}$, then we have the following special cases (RH-Ack) and (LH-Ack);

$$\frac{\&_{j=1}^{m} \beta_j(p) \leq \gamma_j(p)}{\&_{j=1}^{m} \beta_j(\mathbf{0}) \leq \gamma_j(\mathbf{0})} \text{(RH-Ack-0)}$$

$$\frac{\&_{j=1}^{m} \gamma_j(p) \leq \beta_j(p)}{\&_{j=1}^{m} \gamma_j(\mathbf{1}) \leq \beta_j(\mathbf{1})} \text{(LH-Ack-0)}$$

- Phase 4: Translation and Output
  Assuming that it was possible to rewrite an initial system in a form to which one of the Ackermann rules is applicable, else hybrid-ALBA reports failure and terminates, we denote the set of *pure* quasi-inequalities obtained in Phase 3 by *pure* $(\phi \leq \psi)$. Let $ALBA(\phi \leq \psi)$ be the set of quasi-inequalities:

  $$\& \left(pure\left(\phi_i \leq \psi_i\right)\right) \Rightarrow \mathbf{m}_0 \leq \neg \mathbf{n}_0$$

  for each $\phi_i \leq \psi_i \in Preprocess(\phi \leq \psi)$. All the members of $ALBA(\phi \leq \psi)$ are propositional variable free, therefore applying the standard translation to each member of $ALBA(\phi \leq \psi)$ will result in a set of first-order correspondents, that is, one for each member of the set of quasi-inequalities. Let

$$ALBA^{FO}(\phi \leq \psi) = \bigwedge_{1 \leq i \leq n} \forall y_{\mathbf{i}} \forall y_{\mathbf{m}_0} \left[\forall x ST_x \left(pure\left(\phi_i \leq \psi_i\right)\right) \Rightarrow \forall x ST_x \left(\mathbf{m}_0 \leq \mathbf{n}_0\right)\right]$$

The standard translation is applied to every member of $pure\,(\phi \leq \psi)$ and universally quantified overall variables $y_{\mathbf{i}}$ corresponding to occurring nominal $\mathbf{i}$, but with $y_{\mathbf{m}_0}$ corresponding to $\mathbf{m}_0$ left free. The conjunction of the first -order formulas obtained in this way is a local first-order correspondent of $\phi \leq \psi$.

Correctness of hybrid-ALBA

Sahlqvist theory consists of two parts: *correspondence* and *preservation*. This dissertation, however, focuses on the correspondence. This is the idea that will be lifted to many-valued hybrid case. In particular, we want to show that whenever hybrid-ALBA succeeds in eliminating all propositional variables from an inequality, the first-order formula returned is locally equivalent on frames to the inequality. The proof of the following theorem proceeds on complex algebras and for the basic modal logic, proof of the analogous theorem is found on [10], [11] and [12]. The hybrid version is given in [13]. One of the aims of this dissertation is to extend such a theorem to many-valued hybrid logic.

Theorem 1.4.16. If hybrid-ALBA succeeds in reducing an inequality $\phi \leq \psi$ and gives out $ALBA(\phi \leq \psi)$, then

$$\mathfrak{F} \Vdash \phi \leq \psi \text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash \phi \leq \psi \text{ iff } \mathfrak{F} \vDash ALBA^{FO}\left(\phi \leq \psi\right)\left[m_0 = w\right]$$

Proof. The proof is complete when the following chain on equivalences is proven. We will only give the sketch of the proof. The full detailed proof is found in [10]. The strings of equivalences are as follows:

$$\begin{aligned}
\mathfrak{F} \vDash \phi \leq \psi &\text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash Preprocess\left(\phi \leq \psi\right) && (1.3)\\
&\text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash \phi_i \leq \psi_i && (1.4)\\
&\text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash \left(\mathbf{m}_0 \leq \phi_i \& \psi_i \leq \mathbf{n}_0\right) \Rightarrow \left(\mathbf{m}_0 \leq \mathbf{n}_0\right) && (1.5)\\
&\text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash pure\left(\phi_i \leq \psi_i\right) \Rightarrow \left(\mathbf{m}_0 \leq \mathbf{n}_0\right) && (1.6)\\
&\text{ iff } \mathfrak{F}\left(\mathbf{m}_0 = \{w\}\right) \vDash ALBA\left(\phi \leq \psi\right) && (1.7)\\
&\text{ iff } \mathfrak{F} \vDash ALBA^{FO}\left(\phi \leq \psi\right)\left[i_0 = w\right] && (1.8)
\end{aligned}$$

where $\phi_i \leq \psi_i \in Preprocess\left(\phi \leq \psi\right)$. □

Chapter 2

Many-Valued Hybrid Logic

Many-valued hybrid logic was introduced [21] and the accessibility relation between states was also many-valued as in [18]. In [21], the tableau system that is *sound* and *complete* with respect to the semantics was also defined. In this chapter, we will first introduce lattices which will be our truth-value space. Secondly, we will introduce our many-valued hybrid language which will be the language given in [21] and also extend the language by adding variables and operators that will be needed for ALBA algorithm to be given in a later stage of the dissertation. Lastly, we will give out the *duality* between frames and Heyting algebras. This will lead to the introduction of *complex algebras*. Complex algebras are of interest to us because the proof of the correctness of ALBA runs on the complex algebras, and not frames themselves.

2.1 Lattices and Heyting Algebras

This section introduces algebraic structures that will play the role of a *truth-value space* of the many-valued hybrid logic to be introduced later in the dissertation and *complex algebras*, which will play a role in the proof of correctness of ALBA will be algebras of this kind. These algebraic structures will have certain properties that will be very useful to the theory to be developed later in the work. The books by Burris and Sankappanavar [9] and Davey and Priestly [14] are references for the definitions and results in this section.

Definition 2.1.1. An algebra $\mathbf{L} = (L, \vee, \wedge)$ consisting of a nonempty set L together with two binary operations $\vee$ (*join*) and $\wedge$ (*meet*) on L is a *lattice* if it satisfies the following identities for any $x, y, z \in L$:

1. Commutative Laws:
   - $x \vee y = y \vee x$
   - $x \wedge y = y \wedge x$

2. Associative Laws:
   - $x \vee (y \vee z) = (x \vee y) \vee z$
   - $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
3. Idempotent Laws:
   - $x \vee x = x$
   - $x \wedge x = x$
4. Absorption Laws:
   - $x = x \vee (x \wedge y)$
   - $x = x \wedge (x \vee y)$

Additionally, an algebra $\mathbf{L} = (L, \vee, \wedge, 0, 1)$ with two binary operations and two nullary operations 0 (*bottom*) and 1 (*top*) is called a *bounded lattice* if $(L, \vee, \wedge)$ is a lattice and and for all $x \in L$,

1. $x \wedge 0 = 0$
2. $x \vee 1 = 1$

We would like to introduce a second definition of a lattice. But first, we need the notion of a *partial order* on a set.

Definition 2.1.2. A binary relation $\leq$ defined on a nonempty set P is a *partial order* on the set P if the following conditions hold for all $a, b, c \in P$:

1. Reflexivity: $a \leq a$
2. Antisymmetry: If $a \leq b$ and $b \leq a$, then $a = b$
3. Transitivity: If $a \leq b$ and $b \leq c$, then $a \leq c$
   If, in addition, for every a, b in P
4. $a \leq b$ or $b \leq a$

then we say $\leq$ is a *total order* on P. A nonempty set with a partial order defined on it is called a *partially ordered set*, or a *poset*, and if the relation is a total order then the pair $(P, \leq)$ is called a *totally ordered set*, or a *chain*.

Definition 2.1.3. Given a partial order P, we can form a new order P^∂ called the *dual* of P by defining $a \leq b$ to hold in P^∂ if and only if $b \leq a$ holds in P.

Definition 2.1.4. Let A be a subset of a poset P. An element $p \in P$ is an *upper bound* for A if $a \leq p$ for all $a \in A$. An element $p \in P$ is the *least upper bound* or *supremum* (which will be denoted as sup) of A if

1. p is an upper bound of A

2. $a \leq b$ for every a in A implies $p \leq b$ (that is, p is the smallest among the upper bounds of A)

An element $l \in P$ is an *lower bound* for A if $l \leq a$ for all $a \in A$. An element $l \in P$ is the *greatest lower bound* or *infimum* (which will be denoted as inf) of A if

1. l is a lower bound of A

2. $b \leq a$ for every $a \in A$ implies $b \leq l$ (that is, l is the largest among the lower bounds of A)

Definition 2.1.5. Let P be a partially ordered set and $A \subseteq P$.

1. A is a down-set if, whenever $x \in A$, $y \in P$ and $y \leq x$, we have $y \in A$.

2. Dually, A is an up-set if, whenever $x \in A$, $y \in P$ and $y \geq x$, we have $y \in A$.

We now look at the second approach to lattices.

Definition 2.1.6. A poset L is a *lattice* if, and only if, for every $a, b \in L$ both $sup\{a, b\}$ and $inf\{a, b\}$ exist in L.

The following statements are a useful way to show how one can move from one definition of lattice to the other and vice versa.

- If L is a lattice by the first definition, then defining a binary relation $\leq$ on L by $a \leq b$ iff $a = a \wedge b$ yields a partially ordered set.

- If L is a lattice by the second definition, then defining the operations $\vee$ and $\wedge$ by $a \vee b = sup\{a, b\}$ and $a \wedge b = inf\{a, b\}$ yields a lattice since $\vee$ and $\wedge$ satisfies the commutative, associative, idempotent and absorption laws.

Definition 2.1.7. Let L be any lattice. Then a nonempty subset $S \subseteq L$ is called a *sublattice* if S is closed under meet and join.

Example 2.1.8. 1. Consider $(I^+, \leq)$, where I^+ is the set of *positive* integers and $a \leq b$ iff b is a divisor of a. Then for any positive integer n, $(I^n, \leq)$ is a *sublattice* of I^+ where $I^n \subseteq I^+$.

2. Let L be a lattice. Then for any $a \in L$,

$$\uparrow a = \{x \in L \mid a \leq x\}$$

is a sublattice of L which is referred to as a *principle up-set*. Similarly,

$$\downarrow a = \{x \in L \mid x \leq a\}$$

is a sublattice of L which is referred to as a *principle down-set*.

Remark 2.1.9. If L and L' are two lattices such that $L' \subseteq L$, then L' need not be a sublattice of L:
Let S be a group. Consider the following families out subsets of S:

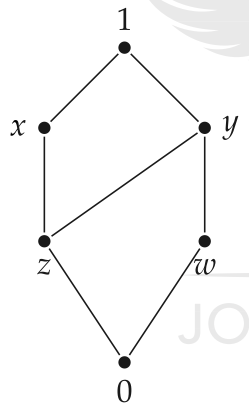
- $\wp(S)$ - *powerset* of S
- $S(G)$ - the collection of all *subgroups* of S

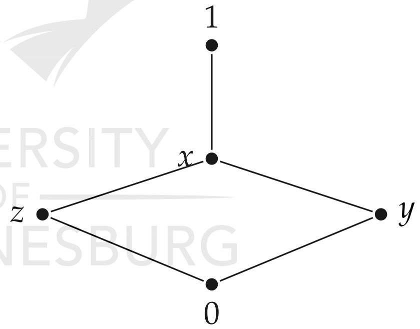
Note that $\wp(S)$ is a lattice with the operations defined by $A \wedge B = A \cap B$ and $A \vee B = A \cup B$. Furthermore, $S(G)$ also forms a lattice under the operations defined by $G_1 \wedge G_2 = G_1 \cap G_2$ and $G_1 \vee G_2$ is the subgroup generated by $G_1 \cup G_2$. Then clearly, $S(G) \subseteq \wp(S)$. But $S(G)$ is not a sublattice of $\wp(S)$ as it is *not closed* under unions, the join operation of $\wp(S)$.

Definition 2.1.10. A *distributive lattice* L is a lattice which satisfies either (and hence both) of the following distributive laws for all $x, y, z \in L$:

1. D1: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
2. D2: $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

Example 2.1.11. Below are the two examples of the bounded distributive lattices:





Lemma 2.1.12. A lattice satisfies D1 iff it satisfies D2.

Proof. We show that D2 follows from D1. Suppose that D1 holds. Then applying it to the right side of D2 and using absorption gives

$$\begin{aligned}(x \vee y) \wedge (x \vee z) &= ((x \vee y) \wedge x) \vee ((x \vee y) \wedge z)\\ &= x \vee ((x \vee y) \wedge z)\\ &= x \vee ((x \wedge z) \vee (y \wedge z))\\ &= x \vee (y \wedge z).\end{aligned}$$

The converse can be shown by applying a dual version of the above argument. □

Note that distributive laws are dual to one another and so a lattice is distributive if and only its dual lattice is distributive.

The two *prototype* non-distributive are the well-known lattices M_3 and N_5:

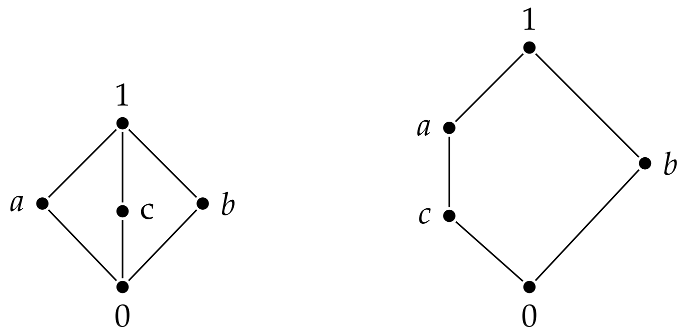


Figure 2.1: The lattices M_3 (left) and N_5 (right)

The following theorem gives us the characteristics that non-distributive lattices have. See proof in [9, Theorem 3.6].

Theorem 2.1.13. A lattice is *non-distributive* if and only if it has M_3 or N_5 as a sublattice.

Definition 2.1.14. A poset P is *complete* if for every subset A of P both $supA$ and $infA$ exist in P. The elements $supA$ and $infA$ will be denoted by $\bigvee A$ and $\bigwedge A$, respectively. All complete posets are lattices, and a lattice L which is complete as a poset is a *complete lattice*.

Infinite analogous of the distributive laws in a complete lattice are the following:

Definition 2.1.15. Given a complete lattice L and an index set I, for $a, b_i \in L$, where $i \in I$ we have that

1. $a \wedge \bigvee_{i\in I} b_i = \bigvee_{i\in I} (a \wedge b_i)$ (join-infinite distributive law)
2. $a \vee \bigwedge_{i\in I} b_i = \bigwedge_{i\in I} (a \vee b_i)$ (meet-infinite distributive law)

The following types of elements play a special role in the representation theory of lattices. *The Fundamental Theorem of Arithmetic* says that every natural number is a *product* of *prime* numbers. Since prime numbers are just the *product-irreducible* natural numbers (other than 1) an analogous result for lattices would state that every element is a *meet* of *meet-irreducible* elements, or dually, a *join* of *join-irreducible* elements. Note that, this will, however, not be true in general but will hold provided we impose an appropriate conditions on the lattice.

Definition 2.1.16. Let L be a lattice.

1. An element $j \in L$ ($j \neq 0$) is *join-irreducible* if j is not the join of two *smaller* elements, that is, if
$$j = a \vee b \text{ then } j = a \text{ or } j = b$$
The set of join-irreducible elements of L is denoted by $J(L)$. An element a is *join-prime* if, for all $b, c \in L$, $a \leq b \vee c$ implies that $a \leq b$ or $a \leq c$.

2. Dually, an element $m \in L$ ($m \neq 1$) is *meet-irreducible* if m is not the meet of two *larger* elements, that is, if
$$m = a \wedge b \text{ then } m = a \text{ or } m = b$$
The set of meet-irreducible elements of L is denoted by $M(L)$. An element a is *meet-prime* if, for all $b, c \in L$, $b \wedge c \leq a$ implies that $b \leq a$ or $c \leq a$.

Every join-prime element is also join-irreducible, and every meet-prime element is also meet-irreducible. If L is a distributive lattice, then the converse also holds.

Remark 2.1.17. Note that in the lattice M_3, elements a, b and c are join-irreducible elements but not are join-prime since $a \leq b \vee c$ but $a \nleq b$ and $a \nleq c$.

Note that infinitary versions of join-irreducible elements and meet-irreducible elements will be important for the algebraic work in Chapter 4. Specifically, these elements will play a huge role in proving the correctness of ALBA.

Definition 2.1.18. Let L be a complete lattice. An element $a \in L$ is *completely join-irreducible* if the following holds:

1. $a \neq 0$.
2. $a = \bigvee S$ implies $a \in S$ for every $S \subseteq L$.

The set of completely join-irreducible elements of L is denoted by $J^\infty(L)$. An element a is *completely join-prime* if, for all $S \subseteq L$, $a \leq \bigvee S$ implies that $a \leq s$ for some $s \in S$. An element $a \in L$ is *completely meet-irreducible* if the following holds:

1. $a \neq 1$.
2. $a = \bigwedge S$ implies $a \in S$ for every $S \subseteq L$.

The set of completely meet-irreducible elements of L is denoted by $M^\infty(L)$. An element a is *completely meet-prime* if, for all $S \subseteq L$, $\bigwedge S \leq a$ implies that $s \leq a$ for some $s \in S$.

The term *perfect* defined below was first used by Dunn, Gehrke and Palmigiano in [15, Definition 2.9].

Definition 2.1.19. A lattice L is perfect if:

1. L is complete.

2. For each $a \in L$, it is the case that $a = \bigvee\{a' \mid a' \in J^{\infty}(L) \text{ and } a' \leq a\}$.

3. For each $a \in L$, it is the case that $a = \bigwedge\{a' \mid a' \in M^{\infty}(L) \text{ and } a \leq a'\}$.

Example 2.1.20. 1. For any set X, the powerset of X is a perfect lattice where join is given by set union and meet is given by set intersection. Then we have that $J^{\infty}(P(X)) = \{\{y\} \mid y \in X\}$ and $M^{\infty}(P(X)) = \{X \backslash \{y\} \mid y \in X\}$.

2. Every *finite lattice* **L** is perfect. This is because L is complete and also the completely join-irreducible elements are simply the join-irreducible elements which are *join-dense* in **L** and the completely meet-irreducible elements are simply the meet-irreducible elements which are *meet-dense* in **L**.

3. Let **L** be a lattice of down-sets of a partially ordered set P. Then the completely join-irreducible elements are the *principle down-sets* and they are join-dense. The completely meet-irreducible elements are the *complements* of principal down-sets and they are meet-dense.

Definition 2.1.21. A Heyting algebra is an algebra $\mathbf{H} = (H, \vee, \wedge, \Rightarrow, 0, 1)$ with three binary and two nullary operations which satisfies:

H1: $(H, \vee, \wedge)$ is a distributive lattice.

H2: $x \wedge 0 = 0$ and $x \vee 1 = 1$

H3: $x \Rightarrow x = 1$

H4: $(x \Rightarrow y) \wedge y = y$ and $x \wedge (x \Rightarrow y) = x \wedge y$

H5: $x \Rightarrow (y \wedge z) = (x \Rightarrow y) \wedge (x \Rightarrow z)$ and $(x \vee y) \Rightarrow z = (x \Rightarrow z) \wedge (y \Rightarrow z)$

The operation $\Rightarrow$ is called the relative pseudo-complement operation on the lattice $(H, \vee, \wedge)$ and is sometimes called the Heyting implication.

Lemma 2.1.22. The operation " $\Rightarrow$ " is order-preserving (or monotonic) in the second coordinate and order-reversing (or antitone) in the first coordinate.

Proof. We show that if $y \leq z$ then $x \Rightarrow y \leq x \Rightarrow z$. So suppose that $y \leq z$. This implies that $y \wedge z = y$. So,

$$\begin{aligned} x \Rightarrow y &= x \Rightarrow (y \wedge z) && \text{(by defintion)} \\ &= (x \Rightarrow y) \wedge (x \Rightarrow z) && \text{(by H5).} \end{aligned}$$

Therefore $x \Rightarrow y \leq x \Rightarrow z$.

For the second part, we show that if $x \leq y$ then $y \Rightarrow z \leq x \Rightarrow z$. So suppose that $x \leq y$. This implies that $x \vee y = y$. So,

$$\begin{aligned} y \Rightarrow z &= (x \vee y) \Rightarrow z && \text{(by definition)} \\ &= (x \Rightarrow z) \wedge (y \Rightarrow z) && \text{(by H5)} \end{aligned}$$

Therefore $y \Rightarrow z \leq x \Rightarrow z$. □

Lemma 2.1.23. If $\mathbf{H} = (H, \vee, \wedge, \Rightarrow, 0, 1)$ is a Heyting algebra and $a, b \in H$ then $a \Rightarrow b$ is the largest element c in H such that $a \wedge c \leq b$.

Proof. We first show that $a \wedge (a \Rightarrow b) \leq b$. So:

$$\begin{aligned} a \wedge c &= a \wedge (a \Rightarrow b) && \text{(by defintion)} \\ &= a \wedge b && \text{(by H4)} \\ &\leq b \end{aligned}$$

Secondly, suppose that there is $d \in H$ such that $a \wedge d \leq b$. We show that $d \leq (a \Rightarrow b)$. Since $\Rightarrow$ is monotone in the second coordinate we have that $a \Rightarrow (a \wedge d) \leq a \Rightarrow b$. So,

$$\begin{aligned} a \Rightarrow (a \wedge d) &= (a \Rightarrow a) \wedge (a \Rightarrow d) && \text{(by H5)} \\ &= 1 \wedge (a \Rightarrow d) && \text{(by H3)} \\ &= (a \Rightarrow b) \end{aligned}$$

Hence we have that $d \leq a \Rightarrow d \leq a \Rightarrow b$ as required. Note that the fact that $d \leq a \Rightarrow d$ follows from H4. Therefore $a \Rightarrow b$ is the largest element in H such that $a \wedge (a \Rightarrow b) \leq b$; that is,

$$a \Rightarrow b = \bigvee \{c : a \wedge c \leq b \text{ and } c \in H\}.$$

□

Example 2.1.24. The following is a perfect Heyting algebra and will be used often in the examples in the text as truth-value space.

1. $\mathbf{K} = \left(\{0, \frac{1}{2}, 1\}, \vee, \wedge, \Rightarrow, 0, 1\right)$ is a perfect Heyting algebra.

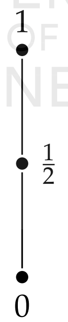


Figure 2.2: The perfect Heyting algebra $\mathbf{K} = \left(\{0, \frac{1}{2}, 1\}, \vee, \wedge, \Rightarrow, 0, 1\right)$

The following lemma shows the properties of the Heyting algebra and these properties will be used (with reference) throughout the proofs of many important theorems in this dissertation.

Lemma 2.1.25. Let H be a Heyting algebra and let $D \subseteq H$ ($D \neq \emptyset$). Then the following hold:

1. $a \Rightarrow (b \Rightarrow c) = a \wedge ((a \wedge b) \Rightarrow (a \wedge c))$.
2. $a \wedge (b \Rightarrow c) = a \wedge ((a \wedge b) \Rightarrow c)$.
3. $a \Rightarrow b = a \Rightarrow (a \wedge b)$.
4. $\bigwedge_{c \in D}(c \Rightarrow a) = (\bigvee_{c \in D} c) \Rightarrow a$.
5. $c \leq a \Rightarrow b$ iff $c \wedge a \leq b$.
6. $c \wedge (a \Rightarrow b) = c \wedge ((c \wedge a) \Rightarrow (c \wedge b))$.
7. if c is a join-irreducible element and $c \leq a \vee b$, then $c \leq a$ or $c \leq b$.

The following lemma and its proof is found in the book by Galatos, Jipsen, Kowalski and Ono [19, Lemma 1.4].

Lemma 2.1.26. 1. The distributive laws holds in every Heyting algebra. In fact, the join-infinite distributive law holds for all existing infinite joins. More precisely, if $\bigvee_{i \in I} b_i$ exists then $\bigvee_{i \in I}(a \wedge b_i)$ exists also and $a \wedge \bigvee_{i \in I} b_i$ is equal to $\bigvee_{i \in I}(a \wedge b_i)$.

2. Conversely, for any complete lattice, if the join-infinite distributive law holds on it, then residuals always exist and hence it is also a Heyting algebra. In particular, every finite distributive lattice is a reduct of a Heyting algebra.

Definition 2.1.27. Given a nonempty set W and a Heyting algebra $\mathbf{H}$, the power algebra $\mathbf{H}^W$ is the algebra $(H^W, \wedge, \vee, \Rightarrow, 0, 1)$ with $H^W = \{f \mid f : W \to H\}$, binary operations $\wedge, \vee$ and $\Rightarrow$ and nullary operations 0 and 1 on H^W defined pointwise on H^W. For example, if $f, g \in H^W$ and $w \in W$ then $(f \wedge g)(w) = f(w) \wedge g(w)$.

Note that 0 and 1 are the constant functions such that $0(w) = 0^H$ and $1(w) = 1^H$ for all $w \in W$.

Definition 2.1.28. Given $\mathbf{H}^W$ with $W = \{w_1, \ldots, w_n\}$, the ordered n-tuple $(a_1, \ldots, a_n)$ represents the function $f \in H^W$ such that $f(w_i) = a_i$ for $1 \leq i \leq n$.

Definition 2.1.29. A Heyting algebra $\mathbf{H} = (H, \wedge, \vee, \Rightarrow, 0, 1)$ is said to be a *perfect Heyting algebra* if $(H, \wedge, \vee, 0, 1)$ is a perfect distributive lattice.

The following proposition is important to the proof of correctness of ALBA as it says the power algebra inherits the perfectness of the Heyting algebra. The proof of this proposition can be found in [7, Proposition 3.42].

Proposition 2.1.30. Let $\mathbf{H} = (H, \vee, \wedge, \Rightarrow, 0, 1)$ be a perfect Heyting algebra. Then $\mathbf{H}^W$ is a perfect Heyting algebra.

We now introduce the relevant preliminaries on *adjoints* and *residuals*, which play a crucial role proving the correctness of ALBA which is one of the theorems we aim to extend in this dissertation and hence obtain the correspondence theory for Many-Valued Hybrid Logic. In this section, $\mathbf{L} = (L, \vee, \wedge, 0, 1)$ and $\mathbf{L}' = (L', \vee', \wedge', 0', 1')$ are complete lattices. Given a lattice $\mathbf{L} = (L, \vee, \wedge, 0, 1)$, the *dual lattice* $(L, \wedge, \vee, 1, 0)$ is denoted as $\mathbf{L}^\partial$.

Definition 2.1.31. The monotone maps $f : \mathbf{L} \to \mathbf{L}'$ and $g : \mathbf{L}' \to \mathbf{L}$ form an *adjoint pair* (notation : $f \dashv g$), if for every $x \in L, y \in L'$,

$$f(x) \leq y \text{ iff } x \leq g(y)$$

Whenever $f \dashv g$, we call f the *left adjoint* of g and g the *right adjoint* of f.

Remark 2.1.32. If a map has a left (or right) adjoint, then the adjoint is *unique* and can be computed *pointwise* from the map itself and the order relation on the lattices. Hence, having a left (or right) adjoint is an intrinsically *order-theoretic* property of maps.

The proof of the following Proposition is found in [19, Lemma 3.3] but it is adapted to the case where f and g are maps between complete lattices.

Proposition 2.1.33. For monotone maps $f : \mathbf{L} \to \mathbf{L}'$ and $g : \mathbf{L}' \to \mathbf{L}$ such that $f \dashv g$, for every $x \in L, y \in L'$,

1. $f(x) = \bigwedge\{y \in L' : x \leq g(y)\}$;
2. $g(y) = \bigvee\{x \in L : f(x) \leq y\}$.

The proof of the following Proposition is found in [19, Lemma 3.6].

Proposition 2.1.34.
1. f is completely join-preserving iff it has a right adjoint;
2. f is completely meet-preserving iff it has a left adjoint.

Definition 2.1.35. The n-ary maps $f : \mathbf{L}^n \to \mathbf{L}$ and $g : \mathbf{L}^n \to \mathbf{L}$ form a *residual pair* in the ith coordinate (notation : $f \dashv_i g$), if for all $x_1, x_2, \ldots x_n \in L$,

$$f(x_1, x_2, \ldots, x_i, \ldots x_n) \leq y \text{ iff } x_i \leq g(x_1, x_2, \ldots, y, \ldots x_n)$$

Whenever $f \dashv_i g$, we call f the *left residual* of g in the ith coordinate and g the *right residual* of f in the ith coordinate.

Note that Proposition 2.1.33 and Proposition 2.1.34 can easily be extended to the case where f and g form a residual pair in the ith coordinate.

Proposition 2.1.36. For n-ary maps $f : \mathbf{L}^n \to \mathbf{L}$ and $g : \mathbf{L}^n \to \mathbf{L}$ such that $f \dashv_i g$, for every $x_1, x_2, \ldots, x_n, y \in L'$,

1. $f(x_1, x_2, \ldots, x_i, \ldots, x_n) = \bigwedge\{y \in L : x_i \leq g(x_1, x_2, \ldots, y, \ldots, x_n)\}$;

2. $g(x_1, x_2, \ldots, y, \ldots, x_n) = \bigvee\{x_i \in L : f(x_1, x_2, \ldots, x_i \ldots, x_n) \leq y\}$.

Proposition 2.1.37. For any n-ary map $f : \mathbf{L}^n \to \mathbf{L}$,

1. f is completely join-preserving in the ith coordinate iff it has a right residual in that same coordinate;

2. f is completely meet-preserving in the ith coordinate iff it has a left residual in that same coordinate.

2.2 Many-Valued Hybrid Logic

We will build our language following the setting in the paper [21] where we would define our models to have a specific function to evaluate nominals in the sense that every nominal will be true only at a unique state. The reason for this is the fact that we want our nominals in the many-valued setting to behave as in the classical case, that is, we still want our nominals to be used as names for states. Our frames in this setting will be mathematically equivalent to the interpretations of many-valued first-order logic.

2.2.1 Syntax of Many-Valued Hybrid Logic

Let a countable infinite set Φ of propositional variables and a countable infinite set Ω of nominals be given. Our truth space will be a perfect Heyting algebra τ.

Definition 2.2.1. The set of formulas in our language L_τ is given by the following grammar:

$$\phi = \mathbf{t} \mid p \mid \mathbf{i} \mid \psi \wedge \gamma \mid \psi \vee \gamma \mid \psi \to \gamma \mid \Diamond\psi \mid \Box\psi \mid @_{\mathbf{i}}\psi$$

where $p \in \Phi$, $\mathbf{i} \in \Omega$ and each $\mathbf{t}$ corresponds to a $t \in \tau$, and $@_{\mathbf{i}}$ is called a satisfaction operator.

We will define $\neg\phi$ as $\neg\phi = \phi \to 0$.

2.2.2 Semantics for Many-Valued Hybrid Logic

Definition 2.2.2. A τ-frame in the language L is a tuple $\mathfrak{F} = (W, R)$ where

1. W is a set of possible states;

2. $R : W \times W \to \tau$ is the many-valued accessibility relation.

Definition 2.2.3. A τ-model in the language L is a tuple $\mathfrak{M} = (W, R, n, V)$ where

1. W is a set of possible state;

2. $R : W \times W \to \tau$ is the accessibility relation;
3. n is a function interpreting the nominals; that is, $n : \Omega \to W$;
4. V, a valuation $V : W \times \Phi \to \tau$ assigns truth values to propositional variables at each state.

The function n assigns to each state a nominal. The reason for such a function is that the many-valued hybrid version gets to behave like the basic hybrid version by allowing nominals to be true at exactly one state.

As in the basic hybrid case, we want to extend our valuation V to assign truth values to all formulas at each world. The set of formulas will be denoted as $FORM_\tau$.

Definition 2.2.4. Given a model $\mathfrak{M} = (W, R, n, V)$, we can extend the valuation V to all formulas in the following inductive way, where $w \in W$:

1. $V(w, \mathbf{t}) = t$ for all $t \in \tau$
2. $V(w, \mathbf{i}) = \begin{cases} 1 & \text{if } n(\mathbf{i}) = w \\ 0 & \text{otherwise} \end{cases}$
3. $V(w, \psi \wedge \gamma) = V(w, \psi) \wedge V(w, \gamma)$
4. $V(w, \psi \vee \gamma) = V(w, \psi) \vee V(w, \gamma)$
5. $V(w, \psi \to \gamma) = V(w, \psi) \Rightarrow V(w, \gamma)$
6. $V(w, \Diamond\psi) = \bigvee_{u \in W} [Rwu \wedge V(u, \psi)]$
7. $V(w, \Box\psi) = \bigwedge_{u \in W} [Rwu \Rightarrow V(u, \psi)]$
8. $V(w, @_\mathbf{i}\psi) = V(n(\mathbf{i}), \psi)$

Remark 2.2.5. Note that we used the same notation V for the extended valuation which assigns every formula a truth value at each state.

Definition 2.2.6. Let $t \in \tau$. A formula ϕ is *t-true* in a model $\mathfrak{M} = (W, R, n, V)$ at $w \in W$ ($\mathfrak{M}, w \Vdash_t \phi$) if $V(w, \phi) \geq t$. A formula ϕ is *t-true* in a model $\mathfrak{M} = (W, R, n, V)$ ($\mathfrak{M} \Vdash_t \phi$) if $\mathfrak{M}, w \Vdash_t \phi$ for all $w \in W$.

Definition 2.2.7. Let $t \in \tau$. A formula ϕ is *t-valid* in a frame $\mathfrak{F} = (W, R)$ at $w \in W$ ($\mathfrak{F}, w \Vdash_t \phi$) if $V(w, \phi) \geq t$ for all valuations V on $\mathfrak{F}$. A formula ϕ is *t-valid* in a frame $\mathfrak{F} = (W, R)$ ($\mathfrak{F} \Vdash_t \phi$) if $\mathfrak{F}, w \Vdash_t \phi$ for all $w \in W$.

Definition 2.2.8. Let $t \in \tau$. A formula ϕ is *t-valid* at $w \in W$ if $V(w, \mathbf{t} \to \phi) = 1$ for all valuations V on $\mathfrak{F}$.

The two definitions of t-validity are equivalent. Hence we have the following

Lemma 2.2.9. Let $t \in \tau$, ϕ a formula, $\mathfrak{F} = (W, R)$ a frame and $w \in W$. Then

$$\mathfrak{F}, w \Vdash_t \phi \text{ iff } \mathfrak{F}, w \Vdash_1 \mathbf{t} \to \phi.$$

2.2.3 The Extended Many-Valued Hybrid Language

The language L_τ is extended to the language L_τ^+ by adding the modalities $\blacklozenge$ and $\blacksquare$, the non-empty sets J-variables (notation: J-VAR) and M-variables (notation: M-VAR), which will be crucial in our MV-hybrid ALBA algorithm to be introduced, and the operators $@^\#$ and $@^\flat$, which will be the left and right adjoint of the operator @, respectively. The reason for requiring adjoint operators is that the MV-hybrid ALBA algorithm runs on the complex algebra of a τ-frame, and not the τ-frame itself henceforth the algorithm uses algebraic rules such as *residuation* and taking adjoints. The sets J-VAR and M-VAR will be used in the First Approximation rule for the MV-Hybrid ALBA algorithm.

Formulas of L_τ^+ are defined inductively as follows:

$$\phi = \mathbf{t} \mid p \mid \mathbf{i} \mid \mathbf{j} \mid \mathbf{m} \mid \psi \wedge \gamma \mid \psi \vee \gamma \mid \psi \to \gamma \mid \Diamond\psi \mid \Box\psi \mid @_\mathbf{i}\psi \mid \blacklozenge\phi \mid \blacksquare\phi \mid @_\mathbf{i}{}^\#\phi \mid @_\mathbf{i}^\flat\phi$$

where $\mathbf{j} \in J$-VAR, $\mathbf{m} \in M$-VAR, $p \in \Phi$, $\mathbf{i} \in \Omega$ and each $\mathbf{t}$ corresponds to a $t \in \tau$.

The definition for the many-valued τ-frames over L_τ^+ remains unchanged.

Definition 2.2.10. The valuation $V : FORM_\tau^+ \times W \to \tau$ is an extension of $V : FORM_\tau \times W \to \tau$ with the following additional clauses:

1. $V(w, \mathbf{j}) \in J^\infty(\tau)$ for exactly one $w \in W$ and $V(u, \mathbf{j}) = 0$ for all $u \neq w$
2. $V(w, \mathbf{m}) \in M^\infty(\tau)$ for exactly one $w \in W$ and $V(u, \mathbf{m}) = 1$ for all $u \neq w$
3. $V(w, \blacklozenge\phi) = \bigvee\{Ruw \wedge^\tau V(w, \phi) \mid u \in W\}$
4. $V(w, \blacksquare\phi) = \bigwedge\{Ruw \Rightarrow V(w, \phi) \mid u \in W\}$
5. $V(w, @_\mathbf{i}^\#\phi) = \begin{cases} 1 & \text{if } n(\mathbf{i}) \neq w \\ \bigwedge\{V(u, \phi) \mid u \in W\} & \text{if } w = n(\mathbf{i}) \end{cases}$
6. $V(w, @_\mathbf{i}^\flat\phi) = \begin{cases} 0 & \text{if } n(\mathbf{i}) \neq w \\ \bigvee\{V(u, \phi) \mid u \in W\} & \text{if } w = n(\mathbf{i}) \end{cases}$

Remark 2.2.11. We will often write $\vee$ and $\wedge$ for the operations on the Heyting algebra τ instead of $\vee^\tau$ and $\wedge^\tau$. It will be clear from the context whether we are referring to the algebra operations or the logical connectives on formulas.

The following proposition shows that we have indeed defined the left and the right adjoint of the operator @.

Proposition 2.2.12. For any τ-frame $\mathfrak{F} = (W, R)$ in the language L_τ^+ and any valuation V on $\mathfrak{F}$, we have that

1. $V(w, @_\mathbf{i}\alpha) \leq V(w, \beta)$ for all $w \in W$ iff $V(w, \alpha) \leq V(w, @_\mathbf{i}^\#\beta)$ for all $w \in W$

2. $V(w, \alpha) \leq V(w, @_{\mathbf{i}}\beta)$ for all $w \in W$ iff $V(w, @^{\flat}_{\mathbf{i}}\alpha) \leq V(w, \beta)$ for all $w \in W$

Proof. 1. Suppose that $V(w, @_{\mathbf{i}}\alpha) \leq V(w, \beta)$ for all $w \in W$. By definition, $V(w, @_{\mathbf{i}}\alpha) = V(n(\mathbf{i}), \alpha) \leq V(w, \beta)$ for all $w \in W$.

- Case 1: $w \neq n(\mathbf{i})$
  $V(w, @^{\#}_{\mathbf{i}}\beta) = 1 \geq V(w, \alpha)$
- Case 2: $w = n(\mathbf{i})$

$$\begin{aligned} V(w, @^{\#}_{\mathbf{i}}\beta) &= \bigwedge\{V(u, \beta) \mid u \in W\} \\ &\geq V(n(\mathbf{i}), \alpha) && \text{(by assumption)} \\ &= V(w, \alpha) \end{aligned}$$

Conversely, suppose that $V(w, \alpha) \leq V(w, @^{\#}_{\mathbf{i}}\beta)$ for all $\in W$. Then

$$\begin{aligned} V(w, @_{\mathbf{i}}\alpha) &= V(n(\mathbf{i}), \alpha) \\ &\leq V(n(\mathbf{i}), @^{\#}_{\mathbf{i}}\beta) && \text{(by assumption)} \\ &= \bigwedge\{V(u, \beta) \mid u \in W\} \\ &\leq V(w, \beta) \end{aligned}$$

2. Suppose that $V(w, \alpha) \leq V(w, @_{\mathbf{i}}\beta)$ for all $w \in W$. Then, by definition, $V(w, \alpha) \leq V(n(\mathbf{i}), \beta) = V(w, @_{\mathbf{i}}\beta)$ for all $w \in W$

- Case 1: $w \neq n(\mathbf{i})$
  $V(w, @^{\flat}_{\mathbf{i}}\alpha) = 0 \leq V(w, \beta)$
- Case 2: $w = n(\mathbf{i})$

$$\begin{aligned} V(w, @^{\flat}_{\mathbf{i}}\alpha) &= \bigvee\{V(u, \alpha) \mid u \in W\} \\ &\leq V(n(\mathbf{i}), \beta) && \text{(by assumption)} \\ &= V(w, \beta) \end{aligned}$$

Conversely, suppose that $V(w, @^{\flat}_{\mathbf{i}}\alpha) \leq V(w, \beta)$ for all $w \in W$. Then

$$\begin{aligned} V(w, @_{\mathbf{i}}\beta) &= V(n(\mathbf{i}), \beta) \\ &\geq V(n(\mathbf{i}), @^{\flat}_{\mathbf{i}}\alpha) && \text{(by assumption)} \\ &= \bigvee\{V(u, \alpha) \mid u \in W\} \\ &\geq V(w, \alpha) \end{aligned}$$

□

2.3 Algebraic Semantics and Duality Between Many-Valued τ-frames and Complex Algebras

Until now we have been focusing on hybrid logic and its relational semantics in terms of frames and models. As is so often the case, these relational structures stand in a close connection to certain types of *algebras* and many constructions and phenomena in the universe of relational structures have analogues in the universe of algebras. For example, bounded morphisms, generated subframes and disjoint unions of relational structures correspond to subalgebras, homomorphisms and products, respectively. The two mathematical universes are systematically related and *duality theory* studies these links. The availability of a duality like this is very useful, as it means that results from one branch of mathematics can be imported into another via duality. In this section we present the duality between τ-frames and a class of perfect Heyting algebras, called *complex algebras*. The duality to be presented will be an extension of the duality between τ-frames and a class of perfect Heyting algebras (called τ-valued modal algebras) which was presented in [7]. Duality is important to us as it is crucial in the proof of the correctness of ALBA [10]. The idea was to prove the correctness of ALBA on complex algebras of frames, and then import those results back to frames via duality. This idea was successfully extended to many-valued modal logic in [7] and we will adapt the same idea to many-valued hybrid logic.

Definition 2.3.1. Given a τ-frame $\mathfrak{F} = (W, R)$, the complex algebra $\mathfrak{F}^+$ of $\mathfrak{F}$ is defined as

$$\mathfrak{F}^+ = (\tau^W, \Diamond_R, \Box_R, @, \{\mathbf{t}\}_{t\in\tau})$$

where

1. τ^W is a power algebra.
2. $\mathbf{t} : W \to \tau$ is a constant function such that $\mathbf{t}(w) = t$ for all $w \in W$
3. The operators $\Diamond_R$ and $\Box_R$ are defined as

   $$(\Diamond_R f)(x) = \bigvee\{Rxy \wedge f(y) \mid y \in W\}$$

   $$(\Box_R f)(x) = \bigwedge\{Rxy \to f(y) \mid y \in W\}$$

   at a state $x \in W$
4. The operator @ (which a binary operator) is defined as

   $$(@_r f)(x) = \bigvee\{r(y) \wedge f(y) \mid y \in W\}$$

   at a state $x \in W$ where
   $X = \{f : W \to \tau \mid f(w) = 1$ for exactly one $w \in W$ and$f(u) = 0$ for all $u \neq w\}$
   and $r \in X$.

FACT: Note that item 4 from Definition 2.3.1 can also be stated as follows:

$$(@_r f)(x) = f(y_o) \text{ where } r(y_o) = 1 \tag{2.1}$$

at a state $x \in W$ where
$X = \{f : W \to \tau \mid f(w) = 1$ for exactly one $w \in W$ and $f(u) = 0$ for all $u \neq w\}$ and $r \in X$.

Lemma 2.3.2. Given any complex algebra $\mathfrak{F}^+$, $\Diamond_R$ is completely join-preserving, $\Box_R$ is completely meet-preserving and @ is both completely join- and meet-preserving in the second coordinate.

Proof. The proofs of $\Diamond_R$ and $\Box_R$ being completely join- and meet-preserving, respectively, are found in [7, Lemma 3.39]. We will show that the @ operator is both completely join- and meet-preserving in the second coordinate. Let $f_i \in \tau^W$ for $i \in I$ and let
$X = \{f : W \to \tau \mid f(w) = 1$ for exactly one $w \in W$ and $f(u) = 0$ for all $u \neq w\}$. Then for all $r \in X$, we have that,

$$\begin{aligned}
\left(@_r \bigvee_i f_i\right)(x) &= \bigvee\{r(y) \wedge \left(\bigvee_i f_i\right)(y) \mid y \in W\} \\
&= \bigvee \bigvee_i \{r(y) \wedge f_i(y) \mid y \in W\} && \text{(By Distribution)} \\
&= \bigvee_i \bigvee \{r(y) \wedge f_i(y) \mid y \in W\} && \text{(by Lemma 2.1.26,1)} \\
&= \bigvee_i (@_r f_i)(x)
\end{aligned}$$

Thus @ is completely join-preserving in the second coordinate. Secondly, we show that @ is completely meet-preserving in the second coordinate. For this proof, we will use the fact we stated in equation 2.1. We consider $r \in X$ where $r(y) = 1$ as the other case where $r(y) = 0$ is trivial true.

$$\begin{aligned}
\left(@_r \bigwedge_i f_i\right)(x) &= \bigvee\{r(y) \wedge \left(\bigwedge_i f_i\right)(y) \mid y \in W\} \\
&= \bigvee\{\left(\bigwedge_i f_i\right)(y) \mid y \in W\} && \text{(By equation 2.1)} \\
&= \left(\bigwedge_i f_i\right)(y) \\
&= \bigwedge_i f_i(y) \\
&= \bigwedge_i (@_r f_i)(x)
\end{aligned}$$

Thus @ is completely meet-preserving in the second coordinate. □

We now define an assignment on complex algebras.

Definition 2.3.3. Given any complex algebra $\mathfrak{F}^+ = (\tau^W, \Diamond_R, \Box_R, @, \{\mathbf{t}\}_{t\in\tau})$, an *assignment* on $\mathfrak{F}^+$ is a map assigning to each propositional variable an element in τ^W, to each nominal an element in X, to each element of J-VAR an element in J and to each element of M-VAR an element in M where

1. $X = \{f : W \to \tau \mid f(w) = 1 \text{ for exactly one } w \in W \text{ and } f(u) = 0 \text{ for all } u \neq w\}$.
2. $J = \{f : W \to \tau \mid f(w) \in J^\infty(\tau) \text{ for exactly one } w \in W \text{ and } f(u) = 0 \text{ for all } u \neq w\}$.
3. $M = \{f : W \to \tau^W \mid f(w) \in M^\infty(\tau) \text{ for exactly one } w \in W \text{ and } f(u) = 1 \text{ for all } u \neq w\}$.

An assignment can be extended to all formulas in the usual inductive way. We now define the notion of a perfect Heyting algebra with operators $\Diamond, \Box$ and @.

Definition 2.3.4. An algebra $(H, \Diamond, \Box, @)$ is called a perfect Heyting algebra with operators $\Diamond, \Box$ and @ if H is a perfect Heyting algebra, $\Diamond$ is completely join-preserving, $\Box$ is completely meet-preserving, and @ is both completely join-preserving and completely meet-preserving in the second coordinate.

Proposition 2.3.5. Given a τ-frame $\mathfrak{F} = (W, R)$, the complex algebra $\mathfrak{F}^+$ is a perfect Heyting algebra with operators.

We now prove a proposition that asserts that the truth of a hybrid formula is equal to a truth-value in a frame if and only if an assignment on a complex algebra of that frame maps the formula to that very same truth-value.

Proposition 2.3.6. Let $W \neq \emptyset$, let $\mathfrak{F} = (W, R)$ be a τ-frame, let ϕ, ψ be hybrid formulas in the language L_τ^+ and let $\mathfrak{F}^+$ be the complex algebra of $\mathfrak{F}$. Then the following holds:

$$\mathfrak{F} \Vdash \phi \leq \psi \text{ iff } \mathfrak{F}^+ \vDash \phi \leq \psi.$$

Proof. We first define an assignment in $\mathfrak{F}^+$ with respect to the valuation and the function that interprets nominals in $\mathfrak{F}$. Let $T = \{\mathbf{t} \mid t \in \tau\}$. Given a valuation V and a function n that interprets nominals in $\mathfrak{F}$, define an assignment $v : \Phi \cup \Omega \cup J\text{-VAR} \cup M\text{-VAR} \cup T \to \mathfrak{F}^+$ by

1. $v(p)(w) = V(w, p)$ for each $p \in \Phi$.
2. $v(\mathbf{i})(w) = \begin{cases} 1 & \text{if } n(\mathbf{i}) = w \\ 0 & \text{otherwise} \end{cases}$
   for each $\mathbf{i} \in \Omega$.
3. $v(\mathbf{j})(w) = V(w, \mathbf{j})$ for each $\mathbf{j} \in J$-VAR.
4. $v(\mathbf{m})(w) = V(w, \mathbf{m})$ for each $\mathbf{m} \in M$-VAR.

5. $v(\mathbf{t})(w) = t$ for each $\mathbf{t} \in T$.

We extend v to all formulas in the usual way. Via the usual induction on formulas we have that $v(\phi)(w) = V(w, \phi)$ for all ϕ. Now it suffices to show that $\mathfrak{F} \nVdash \phi \leq \psi$ iff $\mathfrak{F}^+ \nvDash \phi \leq \psi$. Suppose that $\mathfrak{F} \nVdash \phi \leq \psi$. Then for some $w \in W$ and valuation V, $V(w, \phi) \nleq V(w, \psi)$. Then it follows that $(v(\phi))(w) \nleq (v(\psi))(w)$. For the other direction, suppose that we have an assignment v on a complex algebra $\mathfrak{F}^+$ of a frame $\mathfrak{F}$. Then we can define a valuation V and a function n on $\mathfrak{F}$ in the following way:

1. $V(w, p) = v(p)(w)$ for each $p \in \Phi$.
2. $n(\mathbf{i}) = w$ where $v(\mathbf{i})(w) \neq 0$ for each $\mathbf{i} \in \Omega$.
3. $V(w, \mathbf{j}) = v(\mathbf{j})(w)$ for each $\mathbf{j} \in J$-VAR.
4. $V(w, \mathbf{m}) = v(\mathbf{m})(w)$ for each $\mathbf{m} \in M$-VAR.
5. $V(w, \mathbf{t}) = t$ for each $\mathbf{t} \in T$.

We extend V to all formulas in the usual way. Via the usual induction on formulas we have that $V(w, \phi) = v(\phi)(w)$ for all ϕ. So, if $\mathfrak{F}^+ \nvDash \phi \leq \psi$, then for some $w \in W$ and assignment v on $\mathfrak{F}^+$, we have that $v(\phi)(w) \nleq v(\psi)(w)$. Hence it follows that $V(w, \phi) \neq V(w, \psi)$. $\square$

Chapter 3

Expressivity of Many-Valued Hybrid Logic

Logicians are always interested in the *relations* between different semantic *structures*, and in *operations* that build new structures from old ones. In such research, one would be particularly interested in structural properties that are *preserved* by such relations and operations. Roughly speaking, a property is preserved by a certain relation or operation if, whenever two structures are linked by the relation or operation, then the second structure has the property if the first one has it. We speak of *invariance* if the property is preserved in both directions.

In the next three sections we introduce three important ways of constructing new models from old ones. Two of these constructions will preserves truth of formulas in the states while *disjoint union* construction does not is. We will show that truth is *t-invariant* under *generated submodels* and *bounded morphisms*, but it is not t-invariant under *disjoint unions*. Note that hybrid formulas are also not invariant under disjoint unions in the 2-valued hybrid language.

The theory developed in this section builds on [16]. However, [16] deals with a many-valued modal language, so we extended definitions and theorems in [16] to the many-valued hybrid setting in a natural way by simply accounting for nominals in models of [16].

We first need to generalize the definition of states being equivalent to many-valued setting.

Definition 3.0.1. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be τ-models, $w \in W$ and $w' \in W'$ two states and $t \in \tau$ ($t \neq 0$). We say that w and w' are *t-equivalent* if for every formula ϕ

$$t \wedge V(w, \phi) = t \wedge V'(w', \phi).$$

3.1 Generated Submodels

Generated submodels are a useful way of making smaller models from bigger ones without affecting the information contained in the original model. These models are obtained by throwing away some states in the models without affecting the *satisfaction* of formulas in the remaining states. Moreover, if $\mathfrak{M}'$ is a generated submodel of $\mathfrak{M}$, then $\mathfrak{M}'$ must contain all states of $\mathfrak{M}$ that are named by a nominal $\mathbf{i}$.

The definition of a t-generated submodel for many-valued modal logic is given in [16, Definition 3]. With the combination of the definition of a generated submodel for a classical hybrid language give in [29], we were able to define a t-generated submodel for many-valued hybrid logic as follows:

Definition 3.1.1. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be two τ-models for our language L^τ. Then $\mathfrak{M}'$ is a *t-generated submodel* of $\mathfrak{M}$ (notation: $\mathfrak{M}' \rightarrowtail_t \mathfrak{M}$) if:

1. $W' \subseteq W$;
2. for every $w \in W'$ and $p \in \Phi$, $t \wedge V'(w, p) = t \wedge V(w, p)$;
3. for every $w \in W'$ and $\mathbf{i} \in \Omega$, $n'(\mathbf{i}) = w$ if, and only if, $n(\mathbf{i}) = w$;
4. for states $w, u \in W'$, $t \wedge R'(w, u) = t \wedge R(w, u)$;
5. if $w \in W'$ and $t \wedge R(w, u) \neq 0$, then $u \in W'$.

Note that clause 3 tells us that all states named by a nominal $\mathbf{i}$ in $\mathfrak{M}$ are contained in $\mathfrak{M}'$. Clause 5 corresponds to the closure condition in classical hybrid logic.

The following theorem asserts that hybrid formulas are *t-invariant* under generated submodels. This is also true for a many-valued modal logic (see [16, Theorem 4]) and also true for the classical hybrid logic.

Theorem 3.1.2. For models $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$, let $\mathfrak{M}' \rightarrowtail_t \mathfrak{M}$. Then, for each formula ϕ and state $w \in W'$,

$$t \wedge V(w, \phi) = t \wedge V'(w, \phi).$$

Proof. We proceed by induction on ϕ. If $\phi = \mathbf{s} \in \tau$, then $t \wedge V(w, \mathbf{s}) = t \wedge s$ and $t \wedge V'(w', \mathbf{s}) = t \wedge s$. If $\phi = p$, $p \in \Phi$, then by definition, $t \wedge V(w, p) = t \wedge V'(w, p)$. If $\phi = \mathbf{i}$, $\mathbf{i} \in \Omega$, then we consider two cases:

1. Suppose $n'(\mathbf{i}) = w$. Then by clause 3, $n(\mathbf{i}) = w$ so that $t \wedge V'(w, \mathbf{i}) = t \wedge 1 = t$ and $t \wedge V(w, \mathbf{i}) = t \wedge 1 = t$.
2. Suppose $n'(\mathbf{i}) \neq w$. Again by clause 3 we get that $n(\mathbf{i}) \neq w$ so that $t \wedge V'(w, \mathbf{i}) = t \wedge 0 = 0$ and $t \wedge V(w, \mathbf{i}) = t \wedge 0 = 0$.

Inductive Hypothesis: Assume that for every state $w \in W'$ and formulas ψ and γ we have

$$t \wedge V(w, \psi) = t \wedge V'(w, \psi)$$

and

$$t \wedge V(w, \gamma) = t \wedge V'(w, \gamma).$$

If $\phi = \psi \wedge \gamma$, then

$$\begin{aligned} t \wedge V(w, \psi \wedge \gamma) &= t \wedge (V(w, \psi) \wedge V(w, \gamma)) \\ &= (t \wedge V(w, \psi)) \wedge (t \wedge V(w, \gamma)) \\ &= (t \wedge V'(w, \psi)) \wedge (t \wedge V'(w, \gamma)) \\ &= t \wedge (V'(w, \psi) \wedge V'(w, \gamma)) \\ &= t \wedge V'(w, \psi \wedge \gamma). \end{aligned}$$

If $\phi = \psi \vee \gamma$, then

$$\begin{aligned} t \wedge V(w, \psi \vee \gamma) &= t \wedge (V(w, \psi) \vee V(w, \gamma)) \\ &= (t \wedge V(w, \psi)) \vee (t \wedge V(w, \gamma)) \\ &= (t \wedge V'(w, \psi)) \vee (t \wedge V'(w, \gamma)) \\ &= t \wedge (V'(w, \psi) \vee V'(w, \gamma)) \\ &= t \wedge V'(w, \psi \vee \gamma). \end{aligned}$$

If $\phi = \psi \to \gamma$, then

$$\begin{aligned} t \wedge V(w, \psi \to \gamma) &= t \wedge (V(w, \psi) \Rightarrow V(w, \gamma)) \\ &= t \wedge ((t \wedge V(w, \psi)) \Rightarrow (t \wedge V(w, \gamma))) \qquad \text{(by Lemma 2.1.25,6)} \\ &= t \wedge (((t \wedge V'(w, \psi)) \Rightarrow (t \wedge V'(w, \gamma))) \\ &= t \wedge (V'(w, \psi) \Rightarrow V'(w, \gamma)) \\ &= t \wedge V'(w, \psi \to \gamma). \end{aligned}$$

If $\phi = \Box\psi$, then

$$\begin{aligned}
t \wedge V(w, \Box\psi) &= t \wedge \bigwedge_{u \in W} [R(w,u) \Rightarrow V(u,\psi)] \\
&= \bigwedge_{u \in W} [t \wedge (R(w,u) \Rightarrow V(u,\psi))] \\
&= \bigwedge_{u \in W} [t \wedge (t \wedge R(w,u)) \Rightarrow (t \wedge V(u,\psi))] && \text{(by Lemma 2.1.25,6)} \\
&= \bigwedge_{u \in W'} [t \wedge (t \wedge R(w,u)) \Rightarrow (t \wedge V(u,\psi))] && \text{(by Definition 3.1.1,1)} \\
&= \bigwedge_{u \in W'} [t \wedge (t \wedge R'(w,u)) \Rightarrow (t \wedge V'(u,\psi))] \\
&= \bigwedge_{u \in W'} [t \wedge (R'(w,u) \Rightarrow V'(u,\psi))] \\
&= t \wedge \bigwedge_{u \in W'} [R'(w,u) \Rightarrow V'(u,\psi)] \\
&= t \wedge V'(w, \Box\psi).
\end{aligned}$$

If $\phi = \Diamond\psi$, then

$$\begin{aligned}
t \wedge V(w, \Diamond\psi) &= t \wedge \bigvee_{u \in W} [R(w,u) \wedge V(u,\psi)] \\
&= \bigvee_{u \in W} [t \wedge (R(w,u) \wedge V(u,\psi))] \\
&= \bigvee_{u \in W} [(t \wedge R(w,u)) \wedge (t \wedge V(u,\psi))] \\
&= \bigvee_{u \in W'} [(t \wedge R(w,u)) \wedge (t \wedge V(u,\psi))] && \text{(by Definition 3.1.1,1)} \\
&= \bigvee_{u \in W'} [(t \wedge R'(w,u)) \wedge (t \wedge V'(u,\psi))] \\
&= \bigvee_{u \in W'} [t \wedge (R'(w,u) \wedge V'(u,\psi))] \\
&= t \wedge \bigvee_{u \in W'} [R'(w,u) \wedge V'(u,\psi)] && \text{(by Lemma 2.1.26,1)} \\
&= t \wedge V'(w, \Diamond\psi).
\end{aligned}$$

If $\phi = @_{\mathbf{i}}\psi$, then

$$\begin{aligned}
t \wedge V(w, @_{\mathbf{i}}\psi) &= t \wedge V(n(\mathbf{i}), \psi) \\
&= t \wedge V'(n'(\mathbf{i}), \psi) \\
&= t \wedge V'(w, @_{\mathbf{i}}\psi).
\end{aligned}$$

□

3.2 Bounded Morphisms

We now introduce a *function* f from one model to the other that preserves the satisfaction of the formulas in the states from the *domain* to the *codomain*. To ensure this in the hybrid setting, the function f is designed in such a way that it maps states named by a nominal $\mathbf{i}$ to states named by the same nominal $\mathbf{i}$ on the other model. The definition of a t-bounded morphism for many-valued modal logic is given in [16, Definition 10]. We combined the definition of a bounded morphism for the classical hybrid language [29] and the one for the many-valued modal logic to obtain the following definition for a many-valued hybrid logic:

Definition 3.2.1. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be two τ-models. A *t-bounded morphism* from $\mathfrak{M}$ to $\mathfrak{M}'$ is a function $f : W \to W'$ satisfying the following conditions, where $w, u \in W$, $p \in \Phi$, and $\mathbf{i} \in \Omega$:

1. $t \wedge V(w, p) = t \wedge V'(f(w), p)$;
2. $f(n(\mathbf{i})) = n'(\mathbf{i})$;
3. $t \wedge R(w, u) \leq t \wedge R'(f(w), f(u))$;
4. for every $w \in W$ and $u' \in W'$, if $t \wedge R'(f(w), u') \neq 0$, then there exists $u \in W$ such that $t \wedge R(w, u) = t \wedge R'(f(w), u')$ and $f(u) = u'$.

If f is onto, we call $\mathfrak{M}'$ a *t-bounded morphic image* of $\mathfrak{M}$ through f (notation:$\mathfrak{M} \twoheadrightarrow_t \mathfrak{M}'$).

The clause 4 is the back condition which corresponds to the back condition of the bounded morphism between two classical models.

We now consider an example of a bounded morphism between two τ-models.

Example 3.2.2. For this particular example, we use our truth-value space as the three-element Heyting algebra given in Example 2.1.24,1. Suppose that we have two τ-models $\mathfrak{M} = (W, R, n, V)$ and
$\mathfrak{M}' = (W', R', n', V')$ such that $W = \{w, u, v, s\}$, $R(w, u) = 1, R(u, s) = R(u, v) = \frac{1}{2}$ and the other possible relations between these states are equal to 0. Furthermore, for $p \in \Phi$ we have $V(w, p) = \frac{1}{2}$, $V(u, p) = V(s, p) = V(v, p) = 0$ and $n(\mathbf{i}) = u$ where $\mathbf{i} \in \Omega$. Also, $W' = \{w', u', v'\}, R'(w', u') = 1, R'(u', v') = \frac{1}{2}$ and all other possible relations between these states are equal to 0. Furthermore, for $p \in \Phi$ we have $V'(w', p) = 1$, $V'(u'p) = V'(v', p) = 0$ and $n'(\mathbf{i}) = u'$. Then, mapping the states from $\mathfrak{M}$ to $\mathfrak{M}'$ as:

- $f(w) = w'$

- $f(u) = u'$
- $f(s) = v'$
- $f(v) = v'$

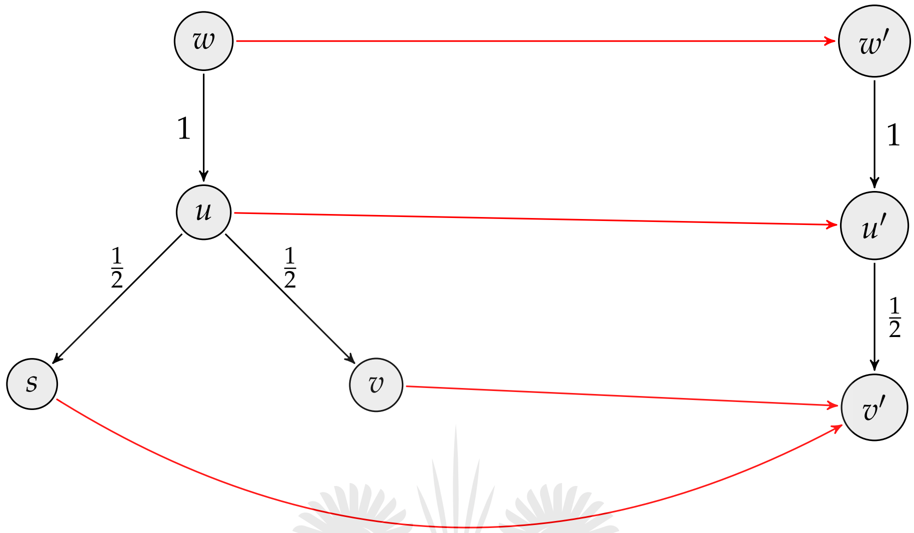


Figure 3.1: Two τ-models with a $\frac{1}{2}$-bounded morphism between them.

yields a $\frac{1}{2}$-bounded morphism between $\mathfrak{M}$ and $\mathfrak{M}'$, but not a 1-bounded morphism. It is easy to check that it is indeed a $\frac{1}{2}$-bounded morphism. We will show why it is not a 1-bounded morphism between $\mathfrak{M}$ and $\mathfrak{M}'$. Suppose, for the sake of a contradiction, that f is a 1-bounded morphism between $\mathfrak{M}$ and $\mathfrak{M}'$. Then it should be the case that $1 \wedge V(w, p) = 1 \wedge V'(w', p)$. But $1 \wedge V(w, p) = 1 \wedge \frac{1}{2} = \frac{1}{2}$ and $1 \wedge V'(w', p) = 1 \wedge 1 = 1$, hence, f cannot be a 1-bounded morphism between $\mathfrak{M}$ and $\mathfrak{M}'$.

We now state and prove the theorem that says that hybrid formulas are t-invariant under t-bounded morphism.

Theorem 3.2.3. Let $\mathfrak{M} = (W, R, n, v)$ and $\mathfrak{M}' = (W', R', n', V')$ be two τ-models and $f : W \to W'$ a t-bounded morphism. Then, for each formula ϕ and state w of $\mathfrak{M}$,

$$t \wedge V(w, \phi) = t \wedge V'(f(w), \phi).$$

Proof. We proceed by induction on ϕ. If $\phi = \mathbf{s} \in \tau$, then $t \wedge V(w, \mathbf{s}) = t \wedge s$ and $t \wedge V'(f(w), \mathbf{s}) = t \wedge s$. If $\phi = p$, $p \in \Phi$, then by definition, $t \wedge V(w, p) = t \wedge V'(f(w), p)$. If $\phi = \mathbf{i}$, $\mathbf{i} \in \Omega$, then we consider two cases:

1. Suppose $n(\mathbf{i}) = w$. Then $n'(\mathbf{i}) = f(w)$ so that $t \wedge V(w, \mathbf{i}) = t \wedge 1 = t$ and $t \wedge V'(f(w), \mathbf{i}) = t \wedge 1 = t$.

2. Suppose $n(\mathbf{i}) \neq w$. Then $n'(\mathbf{i}) \neq f(w)$ so that $t \wedge V(w, \mathbf{i}) = t \wedge 0 = 0$ and $t \wedge V'(f(w), \mathbf{i}) = t \wedge 0 = 0$.

Inductive Hypothesis: Assume that for every state $w \in W$ and formulas ψ and γ we have

$$t \wedge V(w, \psi) = t \wedge V'(f(w), \psi)$$

and

$$t \wedge V(w, \gamma) = t \wedge V'(f(w), \gamma).$$

The cases where $\phi = \psi \wedge \gamma$, $\psi \vee \gamma$, or $\psi \to \gamma$ run the same as in the proof of t-generated submodels, so we skip those.

Suppose $\phi = \Box\psi$. We split our proof into two parts.

1. The first part is to show that $t \wedge V(w, \Box\psi) \geq t \wedge V'(f(w), \Box\psi)$. The details are as follows:

$$\begin{aligned}
t \wedge V(w, \Box\psi) &= t \wedge \bigwedge_{u \in W} [R(w,u) \Rightarrow V(u,\psi)] \\
&= \bigwedge_{u \in W} [t \wedge (R(w,u) \Rightarrow V(u,\psi))] \\
&= \bigwedge_{u \in W} t \wedge [t \wedge R(w,u) \Rightarrow t \wedge V(u,\psi)] && \text{(By Lemma2.1.25, 6)} \\
& && (3.1) \\
&\geq \bigwedge_{u \in W} t \wedge [(t \wedge R'(f(w), f(u)) \Rightarrow t \wedge V'(f(u),\psi))] && (3.2) \\
&= t \wedge \bigwedge_{u \in W} [R'(f(w), f(u)) \Rightarrow V'(f(u),\psi)] && \text{(By Lemma2.1.25, 6)} \\
&= t \wedge \bigwedge_{u' \in W'} [R'(f(w), u') \Rightarrow V'(u',\psi)] && \text{(since } f \text{ is onto)} \\
&= t \wedge V'(f(w), \Box\psi).
\end{aligned}$$

From 3.1 to 3.2 we used the forth condition of f and then applied the inductive hypothesis.

2. For the second inequality $t \wedge V'(f(w), \Box\psi) \geq t \wedge V(w, \Box\psi)$ we will use the back condition and the Axiom of Choice. Here are the details:

We want to find a suitable function $h : W' \to W$. Let

$$X = \{u' \in W' : t \wedge R'(f(w), u') \neq 0\}.$$

By the back condition there exists $u \in W$ such that $t \wedge R'(f(w), u') = t \wedge R(w,u)$ and $f(u) = u\prime$ for each $u' \in X$. Let

$$Y = \{u \in W : u' \in X \text{ and } t \wedge R(w,u) = t \wedge R'(f(w), u') \text{ and } f(u) = u'\}.$$

We know by the back condition that Y is nonempty. Note that $T = \{\langle u', u\rangle : u' \in X, u \in Y\}$ is a relation with dom$T$ = X. By the Axiom of Choice, there is a function $h \subseteq T$ with dom$h$ = X. Note that $h(u') \in f^{-1}[\{u'\}] \subseteq Y \subseteq W$ such that $t \wedge R'(f(w), u') = t \wedge R(w, h(u'))$. Now,

$$
\begin{aligned}
t \wedge V'(f(w), \Box\psi) &= t \wedge \bigwedge_{u' \in W'} [R'(f(w), u') \Rightarrow V'(u', \psi)] \\
&= \bigwedge_{u' \in W'} [t \wedge (R'(f(w), u') \Rightarrow V'(u', \psi))] \\
&= \bigwedge_{u' \in W'} [t \wedge (t \wedge R'(f(w), u') \Rightarrow (t \wedge V'(u', \psi)))] \quad \text{(By Lemma2.1.25, 6)} \\
&= \bigwedge_{u' \in W'} t \wedge [(t \wedge R'(f(w), u')) \Rightarrow (t \wedge V'(u', \psi))] \\
&= t \wedge \bigwedge_{u' \in W'} [t \wedge R'(f(w), u') \Rightarrow t \wedge V(u', \psi)] \qquad (3.3) \\
&= t \wedge \bigwedge_{u' \in X} [t \wedge R'(f(w), u') \Rightarrow t \wedge V'(u', \psi)] \qquad (3.4) \\
&= t \wedge \bigwedge_{u' \in X} [t \wedge R(w, h(u')) \Rightarrow t \wedge V'(h(u'), \psi)] \qquad (3.5) \\
&\geq t \wedge \bigwedge_{u \in W} [t \wedge R(w, u) \Rightarrow t \wedge V(u, \psi)] \qquad \text{(Since } h[X] \subseteq W\text{)} \\
&= \bigwedge_{u \in W} [t \wedge ((t \wedge R(w, u)) \Rightarrow (t \wedge V(u, \psi)))] \qquad \text{(By Lemma2.1.25, 6)} \\
&= t \wedge \bigwedge_{u \in W} [R(w, u) \Rightarrow V(u, \psi)] \\
&= t \wedge v(w, \Box\psi).
\end{aligned}
$$

From 3.3 to 3.4 we used the fact that $X \subseteq W'$ but for any $u' \in W' \backslash X$ such that $t \wedge R'(f(w), u') = 0$, we have that $t \wedge R'(f(w), u') \Rightarrow V'(u', \psi) = 1$. Hence such a u' does not contribute anything in the preceding meet. From 3.4 to 3.5 we used the inductive hypothesis and the definition of h.

Let $\phi = \Diamond\psi$. This is also broken down into two parts. Note that we use the same h that we defined in the $\Box$ case:

1. The proof of the first inequality proceeds as follows:

$$
\begin{aligned}
t \wedge V(w, \Diamond\psi) &= t \wedge \bigvee_{u \in W} [R(w,u) \wedge V(u,\psi)] \\
&= \bigvee_{u \in W} [t \wedge (R(w,u) \wedge V(u,\psi))] \\
&= \bigvee_{u \in W} [(t \wedge R(w,u)) \wedge (t \wedge V(u,\psi))] && (3.6) \\
&\leq \bigvee_{u \in W} [t \wedge R'(f(w), f(u)) \wedge (t \wedge V'(f(u),\psi))] && (3.7) \\
&= \bigvee_{u' \in W'} [t \wedge R'(f(w), u') \wedge (t \wedge V'(u',\psi))] && (\text{Since } f \text{ is onto}) \\
&= t \wedge \bigvee_{u' \in W'} [R'(f(w), u') \wedge V'(u',\psi)] && (\text{ by Lemma 2.1.26,8}) \\
&= t \wedge V'(f(w), \Diamond\psi)
\end{aligned}
$$

From 3.6 to 3.7 we used the forth condition of f and then apply the inductive hypothesis.

2. The second inequality proceeds as follows:

$$
\begin{aligned}
t \wedge V'(f(w), \Diamond\psi) &= t \wedge \bigvee_{u' \in W'} [R'(f(w), u') \wedge V'(u',\psi)] \\
&= \bigvee_{u' \in W'} [t \wedge R'(f(w), u') \wedge (t \wedge V'(u',\psi))] && (3.8) \\
&= \bigvee_{u' \in X} [t \wedge R'(f(w), u') \wedge (t \wedge V'(u',\psi))] && (3.9) \\
&= \bigvee_{u' \in X} [t \wedge R(w, h(u')) \wedge t \wedge V(h(u'),\psi)] && (3.10) \\
&= \bigvee_{u \in Y} [t \wedge R(w,u) \wedge t \wedge V(u,\psi)] && (\text{Since } h[X] = Y) \\
&\leq \bigvee_{u \in W} [t \wedge R(w,u) \wedge t \wedge V(u,\psi)] && (\text{since } Y \subseteq W) \\
&= t \wedge \bigvee_{u \in W} [R(w,u) \wedge V(u,\psi)] && (\text{ by Lemma 2.1.26,8}) \\
&= t \wedge V(w, \Diamond\psi)
\end{aligned}
$$

From (3.8) to (3.9) we used the fact that $X \subseteq W'$ but for any $u' \in W' \backslash X$ such that $t \wedge R'(f(w), u') = 0$ we have that $t \wedge R'(f(w), u') \wedge t \wedge V'(u', \psi) = 0$. Hence such a u' does not contribute anything in the preceding join. From (3.9) to (3.10) we used the inductive hypothesis and the definition of h.

If $\phi = @_{\mathbf{i}}\psi$. Then

$$\begin{aligned} t \wedge V(w, @_{\mathbf{i}}\psi) &= t \wedge V(n(\mathbf{i}), \psi) \\ &= t \wedge V'(n'(\mathbf{i}), \psi) && \text{(By inductive hypothesis)} \\ &= t \wedge V'(f(w), @_{\mathbf{i}}\psi). \end{aligned}$$

□

3.3 Disjoint Unions

Disjoint unions are a useful way of making bigger models from smaller ones in such a way that the bigger models preserves the atomic information of all the small models in one place. This is only achieved in the modal language (both classical and many-valued cases). The disjoint union model is obtained by combining together models that have *disjoint domains*. It gathers together all the information in the smaller models unchanged. Consequently, the relation of states remains the same and the truth of formulas at states remains the same in the disjoint union model. The following definition is for modal logic (specifically, many-valued modal logic) and was defined in [16, Definition 6].

Definition 3.3.1. For an index set K, let $\mathfrak{M}_k = (W_k, R_k, n_k, V_k)(k \in K)$ be a collection of disjoint (for every $a, b \in K$, $W_a \cap W_b$ is empty) models. The *t-disjoint union* of this collection, is the model $\biguplus_k \mathfrak{M}_k = (W, R, n, V)$, where

1. W is the union of the sets W_k;
2. for every $w \in W_k, u \in W_j$: $t \wedge R(w,u) = t \wedge R_k(w,u)$ if $k = j$, else $t \wedge R(w,u) = 0$;
3. for every $w \in W_k$ and $p \in \Phi$, $t \wedge V(w,p) = t \wedge V_k(w,p)$;

Our aim is clearly to extend this definition of a disjoint union to our many-valued hybrid logic so that we can prove the invariance results of hybrid formulas under disjoint unions. However, while doing so, we realized that the resulting structure of taking the union of the sets of worlds is not model since there is no sensible way of defining a nominal interpreter function that would make each nominal true in a unique state.

3.4 Notions of Bisimulation for Many-Valued Hybrid Logic

This section introduces *strong t-bisimulations* which are *relations* between two models. The related states carry the same *atomic information*, that is, for any two models $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$, and $t \in \tau$ ($t \neq 0$), we have that $V(w,p) = V'(w',p)$ where $w \in W$ and $w' \in W'$ are related states and $p \in \Phi$. In addition, whenever it is possible to make a transition in one model, it is possible to make a matching transition in the other model. Our end goal is to obtain the results that say for $t \in \tau$ ($t \neq 0$), t-equivalence implies t-bisimulation and vice versa.

We will also introduce a *weak t-bisimulation*, which is a family of relations between two models indexed by elements of τ subject to some constraints. The purpose of introducing such weak t-bisimulations is to prove that t-equivalence of states implies weak t-bisimulation. This is the main result of this section (see Theorem 3.4.11). This implication fails to go through with the strong t-bisimulation.

The theory in this section is based on the paper [17]. The theory in [17] is based on many-valued modal language. The definitions and theorems in [17] will be extended by inclusion of nominals to make the setting hybrid.

Strong Bisimulation for MVHL

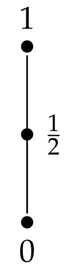
The definition of the strong t-bisimulation in [17] is an analogous to definition of the classical modal bisimulation. However, we add an interpreter function for nominals to the modal strong t-bisimulation to make it a hybrid strong t-bisimulation.

Definition 3.4.1. Given two τ-models $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ and a truth value $t \in \tau$ ($t \neq 0$), a non-empty relation $B \subseteq W \times W'$ is a *strong t-bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$ if for any pair $(w, w') \in B$

1. $t \wedge V(w, p) = t \wedge V'(w', p)$ for every $p \in \Phi$;
2. $n(\mathbf{i}) = w$ if, and only if, $n'(\mathbf{i}) = w'$ for every $\mathbf{i} \in \Omega$;
3. for all $\mathbf{i} \in \Omega$, $(n(\mathbf{i}), n'(\mathbf{i})) \in B$;
4. for every $u \in W$ such that $t \wedge R(w, u) \neq 0$, there exists $u' \in W'$ such that $t \wedge R(w, u) = t \wedge R'(w', u')$ and $(u, u') \in B$ (forth condition);
5. for every $u' \in W'$ such that $= t \wedge R'(w', u') \neq 0$, there exists $u \in W$ such that $t \wedge R(w, u) = t \wedge R'(w', u')$ and $(u, u') \in B$ (back condition).

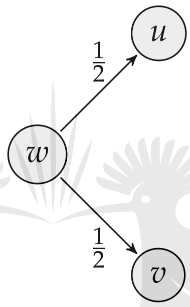
Two states w and w' are called *strongly t-bisimilar* (notation: $w \longleftrightarrow_t w'$ or $\mathfrak{M}, w \longleftrightarrow_t \mathfrak{M}', w'$) if there is a strong t-bisimulation B between $\mathfrak{M}$ and $\mathfrak{M}'$ such that $(w, w') \in B$.

Example 3.4.2. We now consider an example of a strong t-bisimulation between two τ-models. Again we use the 3 element Heyting algebra from Example 2.1.24 as our truth space:

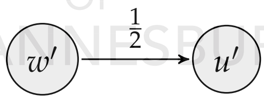


Now, our two τ-models are as follows:

1. $\mathfrak{M} = (\{w, u, v\}, R, n, V)$ is such that $n(\mathbf{i}) = w$, $R(w, u) = R(w, v) = \frac{1}{2}$ and $R(w, w) = R(u, u) = R(v, v) = R(u, v) = R(v, u) = R(u, w) = R(v, w) = 0$. Suppose that $\Phi = \{p\}$ and we have that $V(u, p) = V(v, p) = \frac{1}{2}$ and $V(w, p) = 0$. That is,



2. $\mathfrak{M}' = (\{w', u'\}, R', n', V')$ is such that $n'(\mathbf{i}) = w'$, $R'(w', u') = \frac{1}{2}$ and $R'(w', w') = R'(u', u') = R'(u', w') = 0$. Suppose that $\Phi = \{p\}$ and we have that $V'(u', p) = 1$ and $V'(w', p) = 0$. That is,



The following relation $Z = \{(w, w'), (u, u'), (v, u')\}$ is a $\frac{1}{2}$-bisimulation between these two τ-models but it is not 1-bisimulation between the two models. It is easy to verify the Z is a $\frac{1}{2}$-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. We will show why Z is not a 1-bisimulation. Suppose that Z was a 1-bismulation. Then we must have that $1 \wedge V(w, p) = t \wedge V'(w', p)$ for every $p \in \Phi$ and every $(w, w') \in Z$. But we have that $1 \wedge V(u, p) = 1 \wedge \frac{1}{2} = \frac{1}{2}$ and $1 \wedge V'(u', p) = 1 \wedge 1 = 1$ which is a contradiction. The argument is the same for v and u'. Hence, Z is not a 1-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$.

We now state a basic theorem, which asserts that t-strong bisimulation implies t-equivalence. This is given for the many-valued modal logic in [17, Theorem 3.3] without its proof. We will give the full details of the proof, with the extended hybrid machinery.

Theorem 3.4.3. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be τ-models, $w \in W$ and $w' \in W'$ two states and $t \in \tau (t \neq 0)$. If $\mathfrak{M}, w \longleftrightarrow_t \mathfrak{M}', w'$, then $t \wedge V(w, \phi) = t \wedge V'(w', \phi)$ for every formula ϕ.

Proof. We proceed by induction on ϕ. If $\phi = \mathbf{s}$, $\mathbf{s} \in \tau$, then $t \wedge V(w, \mathbf{s}) = t \wedge s$ and $t \wedge V'(w', \mathbf{s}) = t \wedge s$. If $\phi = p$, $p \in \Phi$, then by definition $t \wedge V(w, p) = t \wedge V'(w', p)$. If $\phi = \mathbf{i}$, $\mathbf{i} \in \Omega$, then we consider two cases:

1. Suppose $n(\mathbf{i}) = w$. Then by definition of the strong bisimulation $n'(\mathbf{i}) = w'$. So, $t \wedge V(w, \mathbf{i}) = t \wedge 1 = t$ and $t \wedge V'(w', \mathbf{i}) = t \wedge 1 = t$.

2. Suppose $n(\mathbf{i}) \neq w$. Then by definition of the strong bisimulation $n'(\mathbf{i}) \neq w'$. So $t \wedge V(w, \mathbf{i}) = t \wedge 0 = 0$ and $t \wedge V'(w', \mathbf{i}) = t \wedge 0 = 0$.

Inductive Hypothesis: Assume that for all states $w \in W$, $w' \in W'$, all $t \in \tau \backslash \{0\}$ and formulas ψ and γ, if $w \longleftrightarrow_t w'$ then

$$t \wedge V(w, \psi) = t \wedge V'(w', \psi)$$

and

$$t \wedge V(w, \gamma) = t \wedge V'(w', \gamma).$$

If $\phi = \psi \wedge \gamma$, then

$$\begin{aligned} t \wedge V(w, \psi \wedge \gamma) &= t \wedge (V(w, \psi) \wedge V(w, \gamma)) \\ &= (t \wedge V(w, \psi)) \wedge (t \wedge V(w, \gamma)) \\ &= (t \wedge V'(w', \psi)) \wedge (t \wedge V'(w', \gamma)) \quad \text{(By inductive hypothesis)} \\ &= t \wedge (V'(w', \psi) \wedge V'(w', \gamma)) \\ &= t \wedge V'(w', \psi \wedge \gamma). \end{aligned}$$

If $\phi = \psi \vee \gamma$, then

$$\begin{aligned} t \wedge V(w, \psi \vee \gamma) &= t \wedge (V(w, \psi) \vee V(w, \gamma)) \\ &= (t \wedge V(w, \psi)) \vee (t \wedge V(w, \gamma)) \\ &= (t \wedge V'(w', \psi)) \vee (t \wedge V'(w', \gamma)) \quad \text{(By inductive hypothesis)} \\ &= t \wedge (V'(w', \psi) \vee V'(w', \gamma)) \\ &= t \wedge V'(w', \psi \vee \gamma). \end{aligned}$$

If $\phi = \psi \to \gamma$, then

$$\begin{aligned}
t \wedge V(w, \psi \to \gamma) &= t \wedge (V(w,\psi) \Rightarrow V(w,\gamma)) \\
&= t \wedge ((t \wedge V(w,\psi)) \Rightarrow (t \wedge V(w,\gamma))) && \text{(Lemma 2.1.25, 6)} \\
&= t \wedge ((t \wedge V'(w',\psi)) \Rightarrow (t \wedge V'(w',\gamma))) && \text{(By inductive hypothesis)} \\
&= t \wedge (V'(w',\psi) \Rightarrow V'(w',\gamma)) && \text{(Lemma 2.1.25, 6)} \\
&= t \wedge V'(w', \psi \to \gamma).
\end{aligned}$$

Suppose $\phi = \Box\psi$. Let $W_t = \{u \in W : t \wedge R(w,u) \neq 0\}$ and let $W'_t = \{u' \in W' : R'(w',u') \neq 0\}$. Now

$$\begin{aligned}
t \wedge V(w, \Box\psi) &= t \wedge \bigwedge_{u \in W} [R(w,u) \Rightarrow V(u,\psi)] \\
&= \bigwedge_{u \in W} [t \wedge (R(w,u) \Rightarrow V(u,\psi))] \\
&= \bigwedge_{u \in W} t \wedge [(t \wedge R(w,u)) \Rightarrow (t \wedge V(u,\psi))] && \text{(Lemma2.1.25, 6)} \\
& && (3.11) \\
&= \bigwedge_{u \in W_t} t \wedge [(t \wedge R(w,u)) \Rightarrow (t \wedge V(u,\psi))] && (3.12) \\
&= \bigwedge_{u' \in W'_t} t \wedge \left[(t \wedge R'(w',u')) \Rightarrow (t \wedge V'(u',\psi))\right] && (3.13) \\
&= \bigwedge_{u' \in W'} t \wedge \left[(t \wedge R'(w',u')) \Rightarrow (t \wedge V'(u',\psi))\right] && (3.14) \\
&= \bigwedge_{u' \in W'} \left[t \wedge (R'(w',u') \Rightarrow V'(u',\psi))\right] && \text{(Lemma2.1.25, 6)} \\
&= t \wedge \bigwedge_{u' \in W'} \left[R'(w',u') \Rightarrow V'(u',\psi)\right] \\
&= t \wedge V'(w', \Box\psi).
\end{aligned}$$

From 3.11 to 3.12 we used the fact that for $u \in W \backslash W_t$ such that $t \wedge R(w,u) = 0$ we have that $t \wedge R(w,u) \Rightarrow t \wedge V(u,\psi) = 1$. Hence such a u does not contribute anything in the preceding meet. From 3.12 to 3.13 we made use of the forth condition and the back condition of the bisimulation to get equality, the inductive hypothesis and the definition of W'_t. From 3.13 to 3.14 we used the same fact we used from 3.11 to 3.12 (in the opposite direction).

Suppose that $\phi = \Diamond\psi$. Note that we use the same W_t and W'_t that were defined in the box case.

$$
\begin{aligned}
t \wedge V(w, \Diamond\psi) &= t \wedge \bigvee_{u \in W} [R(w,u) \wedge V(w,u)] \\
&= \bigvee_{u \in W} [t \wedge (R(w,u) \wedge V(w,u))] \\
&= \bigvee_{u \in W} [(t \wedge R(w,u)) \wedge (t \wedge V(w,u))] && (3.15) \\
&= \bigvee_{u \in W_t} [(t \wedge R(w,u)) \wedge (t \wedge V(w,u))] && (3.16) \\
&= \bigvee_{u' \in W'_t} \left[(t \wedge R'(w',u')) \wedge (t \wedge V'(w',u'))\right] && (3.17) \\
&= \bigvee_{u' \in W'} \left[(t \wedge R'(w',u')) \wedge (t \wedge V'(w',u'))\right] && (3.18) \\
&= \bigvee_{u' \in W'} \left[t \wedge (R'(w',u') \wedge V'(u',\psi))\right] \\
&= t \wedge \bigvee_{u' \in W'} \left[R'(w',u') \wedge V'(u',\psi)\right].
\end{aligned}
$$

From 3.15 to 3.16 we used the fact that for $u \in W \backslash W_t$ such that $t \wedge R(w,u) = 0$ we have that $t \wedge R(w,u) \wedge t \wedge V(u,\psi) = 0$. Hence such a u does not contribute anything in the preceding join. From 3.16 to 3.17 we used the definition of W'_t, forth condition and back condition of the bisimulation to get equality and the inductive hypothesis. From 3.17 to 3.18 we used the same fact that we used to get from 3.15 to 3.16 but in the opposite direction.

If $\phi = @_{\mathbf{i}}\psi$, then

$$
\begin{aligned}
t \wedge V(w, @_{\mathbf{i}}\psi) &= t \wedge V(n(\mathbf{i}), \psi) \\
&= t \wedge V'(n'(\mathbf{i}), \psi) \quad \text{(Since } (n(\mathbf{i}), n'(\mathbf{i})) \in B \text{ and by Inductive Hypothesis)} \\
&= t \wedge V'(w', @_{\mathbf{i}}\psi).
\end{aligned}
$$

□

We will not be able to prove this converse implication on strong t-bisimulations. Hence, for that matter, we introduce the notion of a weak t-bisimulation. Furthermore, the converse implication will be proved on a special set of models, namely the image-finite models.

Weak Bisimulation for MVHL

Weak bisimulations were introduced in [17] for many-valued modal logic. We extend it the same way we did with the strong t-bisimulation in the previous section to obtained these bisimulations in the hybrid setting. We are still working with a perfect Heyting algebra τ as our truth-value space. Let $J(\tau)$ denote the set of join-irreducible elements of τ. Define the function $D_\tau : \tau\backslash\{0\} \to 2^{J(\tau)}$ by

$$D_\tau(t) = \{c \in J(\tau) : c \leq t\}.$$

Definition 3.4.4. Given two τ-models $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$, a function $Z : \tau\backslash\{0\} \to 2^{W \times W'}$ is a *weak bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$ if it satisfies the following properties:

1. For every $t_1, t_2 \in \tau\backslash\{0\},\ Z(t_1 \vee t_2) = Z(t_1) \cap Z(t_2)$ (consistency).
2. For every join-irreducible element $t \in J(\tau)$ and any pair $(w, w') \in Z(t)$
   (a) $t \wedge V(w, p) = t \wedge V'(w', p)$ for every $p \in \Phi$;
   (b) for every $u \in W$ such that $t \wedge R(w, u) \neq 0$ and for every $c \in D_\tau(t \wedge R(w, u))$, there exists $u' \in W'$ such that $c \leq R'(w', u')$ and $(u, u') \in Z(c)$ (forth condition);
   (c) for every $u' \in W'$ such that $t \wedge R'(w', u') \neq 0$ and for every $c \in D_\tau(t \wedge R'(w', u'))$, there exists $u \in W$ such that $c \leq R(w, u)$ and $(u, u') \in Z(c)$ (back condition).
3. (a) If $n(\mathbf{i}) = v$ and $n'(\mathbf{i}) = v'$ for every $v \in W$, $v' \in W'$, and $\mathbf{i} \in \Omega$, then $(v, v') \in Z(t)$ for every $t \in \tau$ $(t \neq 0)$.
   (b) $(n(\mathbf{i}), n'(\mathbf{i})) \in Z(t)$ for every $\mathbf{i} \in \Omega$ and every $t \in \tau$ $(t \neq 0)$.

Two states w and w' are called *weakly t-bisimilar* (notation: $w \leftrightsquigarrow_t w'$ or $\mathfrak{M}, w \leftrightsquigarrow_t \mathfrak{M}', w'$) if there is a weak bisimulation Z between $\mathfrak{M}$ and $\mathfrak{M}'$ such that $(w, w') \in Z(t)$.

We now show that we have indeed defined a weaker notion of a bisimulation. The following lemma shows that strong t-bisimulation implies weak t-bisimulation and Example 3.4.6 will show, however, that the converse does not hold in general. The following lemma is given in [17, Lemma 3.9] for many-valued modal logic. Here we extend the lemma to the many-valued hybrid case.

Lemma 3.4.5. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be two τ-models for MVHL, $w \in W$ and $w' \in W'$ two states and $t \in \tau$ $(t \neq 0)$. Then $\mathfrak{M}, w \longleftrightarrow_t \mathfrak{M}', w'$ implies $\mathfrak{M}, w \leftrightsquigarrow_t \mathfrak{M}', w'$.

Proof. Suppose that B is a strong t-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ such that $(w, w') \in B$. Define $Z : \tau\backslash\{0\} \to 2^{W \times W'}$ as follows:

$$Z(a) = \begin{cases} B & \text{if } a \leq t \\ \varnothing & \text{otherwise} \end{cases}$$

We show that Z is a weak bisimulation. For the consistency condition it suffices to show that $Z(a \vee b) = B$ iff $Z(a) \cap Z(b) = B$. That is,

$$\begin{aligned} Z(a \vee b) = B &\text{ iff } a \vee b \leq t \\ &\text{ iff } a \leq t \text{ and } b \leq t \\ &\text{ iff } Z(a) = B \text{ and } Z(b) = B \\ &\text{ iff } Z(a) \cap Z(b) = B. \end{aligned}$$

For the following properties, suppose that $(w, w') \in Z(a)$ for some join-irreducible element $a \in \tau \backslash \{0\}$. By definition of Z we have that $a \leq t$ and hence $a \wedge t = a$.

1. By definition of Z, (w, w') also belongs to B. Hence by the clause 1 of a strong bisimulation (Definition 3.4.1) we have that $t \wedge V(w, p) = t \wedge V'(w', p)$ for every $p \in \Phi$. Hence $a \wedge (t \wedge V(w, p)) = a \wedge (t \wedge V'(w', p))$ which implies that $a \wedge V(w, p) = a \wedge V'(w', p)$.

2. (Back condition) Suppose that $a \wedge R'(w', u') \neq 0$ for some $u' \in W'$ and consider any $c \in D_\tau(a \wedge R'(w', u'))$. Since $a \leq t$, we have that $t \wedge R'(w', u') \neq 0$. Therefore by the back condition of B there exists $u \in W$ such that $t \wedge R(w, u) = t \wedge R'(w', u')$ and $(u, u') \in B$. Also, $c \leq a \wedge R'(w', u') \leq t \wedge R'(w', u') = t \wedge R(w, u) \leq R(w, u)$. Also $c \leq t$ implies that $(u, u') \in Z(c)$. Thus Z satisfies the back condition.

   The forth condition is symmetric to the back condition so we will skip that.

The remaining two clauses do not necessarily depend on a being join-irreducible. The following holds on every non-zero truth value of τ.

1. Suppose that $n(\mathbf{i}) = v$ and $n'(\mathbf{i}) = v'$ for every $v \in W, v' \in W'$ and every $\mathbf{i} \in \Omega$. Then by clause 2 of the strong bisimulation (Definition 3.4.1) we have that $(v, v') \in B$. If $a \leq t$, then $(v, v') \in Z(a)$. If $a \not\leq t$, then $Z(a) = \emptyset$ and we are done.

2. Suppose that $n(\mathbf{i}) = v$ and $n'(\mathbf{i}) = v'$ for every $v \in W, v' \in W'$ and every $\mathbf{i} \in \Omega$. Then by clause 3 of the strong bisimulation (Definition 3.4.1) we have that $(v, v') \in B$. If $a \leq t$, then $(v, v') \in Z(a)$. If $a \not\leq t$, then $Z(a) = \emptyset$ and we are done.

Hence Z is a weak t-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. □

The converse, however, is not true. The converse does hold if our truth space only has two elements. Hence strong bisimulation and weak bisimulation coincide in the classical case. The following example shows that a weak t-bisimulation will not always be a strong t-bisimulation. Note that this example is found in [17] as a many-valued modal example, so we make nominals true at some states to make it a many-valued hybrid example and we will also see how nominals being true at certain states can affect the function not being a weak bisimulation between models.

Example 3.4.6. Let $\Phi = \{p\}$, $\Omega = \{\mathbf{i}\}$, and let τ have at least three elements. Let c be an element in τ that is different from 0 and 1. Let $\mathfrak{M} = (W, R, n, V)$ be a model where $W = \{w, u\}$, $R(w, u) = c$, $R(u, w) = R(w, w) = R(u, u) = 0$, $V(w, p) = 0$, $V(u, p) = c$, and $V(w, \mathbf{i}) = 1$, hence $n(\mathbf{i}) = w$. Also, let $\mathfrak{M}' = (W', R', n', V')$ be a model where $W' = \{w', u'\}$, $R'(w', u') = c$, $R'(u', w') = R'(w', w') = R'(u', u') = 0$, $V'(w', p) = 0$, $V'(u', p) = 1$, and $V'(w', \mathbf{i}) = 1$, hence $n'(\mathbf{i}) = w'$. Define

$$Z(a) = \begin{cases} \{(w, w'), (u, u')\} & \text{if } a \leq c \\ \{(w, w')\} & \text{otherwise} \end{cases}$$

We first show that Z is a weak bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. We will prove this for $a \in \tau$ with $a \leq c$.

1. Consistency: Let $t_1, t_2 \in \tau \backslash \{0\}$ with $t_1 \leq c$ and $t_2 \leq c$. We want to show that $Z(t_1 \vee t_2) = Z(t_1) \cap Z(t_2)$. By definition of Z, $Z(t_1) \cap Z(t_2) = \{(w, w'), (u, u')\}$. So it suffices to show that $t_1 \vee t_2 \leq c$. Since $t_1 \leq c$ and $t_2 \leq c$, it follows from the definition of join that $t_1 \vee t_2 \leq c$.

2. Clearly, $a \wedge V(w, p) = a \wedge 0 = 0$ and $a \wedge V'(w', p) = a \wedge 0 = 0$. Also, $a \wedge V(u, p) = a \wedge c = a$ since $a \leq c$ and $a \wedge V'(u', p) = a \wedge 1 = a$. If $a \not\leq c$, then $(u, u') \notin Z(a)$ and we are done.

3. Clearly, $(n(\mathbf{i}), n'(\mathbf{i})) \in Z(a)$.

4. Forth and Back conditions follow from the fact $R(w, u) = R'(w', u') = c$.

Therefore Z is a weak bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. That is, $\mathfrak{M}, w \leftrightsquigarrow_1 \mathfrak{M}', w'$. Suppose now, for the sake of a contradiction, that $\mathfrak{M}, w \longleftrightarrow_1 \mathfrak{M}', w'$ and let B be a 1-bisimulation such that $(w, w') \in B$. The forth condition implies that $(u, u') \in B$. But now, $1 \wedge V(u, p) = 1 \wedge c = c$ and $1 \wedge V'(u', p) = 1 \wedge 1 = 1$. Hence $1 \wedge V(u, p) \neq 1 \wedge V'(u', p)$. Therefore B violates the base condition, and so, we get a contradiction. Consequently, w and w' are weakly 1-bisimilar but not strongly 1-bisimilar.

The following remark is based on the preceding example.

Remark 3.4.7. Note that if for some nominal $\mathbf{i} \in \Omega$, we let $n(\mathbf{i}) = u$ and $n'(\mathbf{i}) = u'$, then we no longer get a weak bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. This is because, by definition, we must always have that $(n(\mathbf{i}), n'(\mathbf{i})) \in Z(a)$ for every $\mathbf{i} \in \Omega$ but we can see by definition of Z, if $a \not\leq c$, then $Z(a) = \{(w, w')\}$.

Weak Bisimulation and truth invariance

As with the strong t-bisimulation, the weak t-bisimulation implies t-equivalence between related states. The proof is given for many-valued modal logic in [17, Theorem 3.11]. We will extend this by accounting for nominals to make it hybrid.

Theorem 3.4.8. Let $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be models, $w \in W$ and $w' \in W'$ two states and $t \in \tau$, $(t \neq 0)$. If $\mathfrak{M}, w \leftrightsquigarrow_t \mathfrak{M}', w'$ then $t \wedge V(w, \phi) = t \wedge V'(w', \phi)$ for every formula ϕ.

Proof. The proof proceeds by induction on ϕ. We first show the theorem holds for the case where t is join-irreducible. The base cases where $\phi = \mathbf{s}$ or $\phi = p$ or $\phi = \mathbf{i}$, where $s \in \tau$, $p \in \Phi$ and $\mathbf{i} \in \Omega$ run exactly as the proof of Theorem 3.4.3.

Inductive Hypothesis: Assume that for all states $w \in W$, $w' \in W'$, for all join-irreducibles t and formulas ψ and γ, if $w \in W$ and $w' \in W'$ are linked by $Z(t)$ then we have

$$t \wedge V(w, \psi) = t \wedge V'(w', \psi)$$

and

$$t \wedge V(w, \gamma) = t \wedge V'(w', \gamma).$$

The cases where $\phi = \psi \wedge \gamma, \psi \vee \gamma$ or $\psi \rightarrow \gamma$ run exactly as in the proof of Theorem 3.4.3.

Suppose that $\phi = \Box\psi$. The proof splits into two inequalities. Let

$W_t = \{u \in W : t \wedge R(w,u) \neq 0\}$ and $W'_t = \{u' \in W' : t \wedge R'(w',u') \neq 0\}$. Then

$$
\begin{aligned}
t \wedge V(w, \Box\psi) &= t \wedge \bigwedge_{u \in W} [R(w,u) \Rightarrow V(u,\psi)] \\
&= t \wedge \bigwedge_{u \in W} [t \wedge (R(w,u) \Rightarrow V(u,\psi))] \\
&= t \wedge \bigwedge_{u \in W} [t \wedge ((t \wedge R(w,u)) \Rightarrow V(u,\psi))] && \text{(By Lemma2.1.25,2)} \\
&= t \wedge \bigwedge_{u \in W} [(t \wedge R(w,u)) \Rightarrow V(u,\psi)] && (3.19) \\
&= t \wedge \bigwedge_{u \in W_t} [(t \wedge R(w,u)) \Rightarrow V(u,\psi)] && (3.20) \\
&= t \wedge \bigwedge_{u \in W_t} \left[\Big(\bigvee_{c \in D_\tau(t \wedge R(w,u))} c \Big) \Rightarrow V(u,\psi) \right] && \text{(Since } \tau \text{ is perfect} \\
&= t \wedge \bigwedge_{u \in W_t} \bigwedge_{c \in D_\tau(t \wedge R(w,u))} (c \Rightarrow V(u,\psi)) && \text{(By Definition ((2.1.21),H5))} \\
&= t \wedge \bigwedge_{u \in W_t} \bigwedge_{c \in D_\tau(t \wedge R(w,u))} (c \Rightarrow (c \wedge V(u,\psi))) && \text{(By Lemma2.1.25,3)} \\
& && (3.21) \\
&= t \wedge \bigwedge_{u' \in W'_t} \bigwedge_{c \in D_\tau(t \wedge R'(w',u'))} (c \Rightarrow (c \wedge V'(u',\psi))) && (3.22) \\
&= t \wedge \bigwedge_{u' \in W'_t} \bigwedge_{c \in D_\tau(t \wedge R'(w',u'))} (c \Rightarrow V'(u',\psi)) && \text{(By Lemma2.1.25,3)} \\
&= t \wedge \bigwedge_{u' \in W'_t} \left[\Big(\bigvee_{c \in D_\tau(t \wedge R'(w',u'))} c \Big) \Rightarrow V'(u',\psi) \right] && \text{(By Lemma2.1.25,4)} \\
& && (3.23) \\
&\geq t \wedge \bigwedge_{u' \in W'_t} [(R'(w',u') \Rightarrow V'(u',\psi))] && (3.24) \\
&= t \wedge \bigwedge_{u' \in W'} [R'(w',u') \Rightarrow V'(u',\psi)] && (3.25) \\
&= t \wedge V'(w', \Box\psi).
\end{aligned}
$$

From (3.19) to (3.20) we used the fact that for $u \in W \backslash W_t$, $t \wedge R(w,u) = 0$ and so $t \wedge R(w,u) \Rightarrow V(u,\psi) = 1$, hence such a u does not contribute anything in the preceding meet. Line (3.22) follows from (3.21) by the definition of W'_t and for any pair (u,c) in the (3.21), the forth condition gives a $u' \in W'$ such that $c \leq R'(w',u')$ and $(u,u') \in Z(c)$. By Inductive Hypothesis, $c \wedge V(u,\psi) = c \wedge V'(u',\psi)$ and $c \leq t \wedge R(w,u)$

implies $c \leq t \wedge R'(w', u')$. From (3.23) to (3.24) we used the fact that

$$\bigvee_{c \in D_\tau(t \wedge R(w,u))} \leq t \wedge R(w,u) = t \wedge R'(w', u') \leq R'(w', u')$$

and the fact that if $a \leq b$ then $b \Rightarrow c \leq a \Rightarrow c$ for every $a, b, c \in \tau$. From (3.24) to (3.25) we used the fact that for $u' \in W' \backslash W'_t$, $t \wedge R'(w', u') = 0$ and so $t \wedge R'(w', u') \Rightarrow V'(u', \psi) = 1$, hence such u' does not contribute anything in the preceding meet.

The proof of the inequality $t \wedge V'(w', \Box\psi) \leq t \wedge V(w, \Box\psi)$ is symmetrical to the above one so we skip it.

Suppose that $\phi = \Diamond\psi$. Then

$$\begin{aligned}
t \wedge V(w, \Diamond\psi) &= t \wedge \bigvee_{u \in W} [R(w,u) \wedge V(u,\psi)] \\
&= t \wedge \bigvee_{u \in W} [t \wedge (R(w,u) \wedge V(u,\psi))] \\
&= t \wedge \bigvee_{u \in W} [(t \wedge R(w,u)) \wedge V(u,\psi)] && (3.26) \\
&= t \wedge \bigvee_{u \in W_t} [(t \wedge R(w,u)) \wedge V(u,\psi)] && (3.27) \\
&= t \wedge \bigvee_{u \in W_t} \left[\Big(\bigvee_{c \in D_\tau(t \wedge R(w,u))} c \Big) \wedge V(u,\psi) \right] && \text{(Since } \tau \text{ is perfect)} \\
&= t \wedge \bigvee_{u \in W_t} \bigvee_{c \in D_\tau(t \wedge R(w,u))} (c \wedge V(u,\psi)) && (3.28) \\
&= t \wedge \bigvee_{u' \in W'_t} \bigvee_{c \in D_\tau(t \wedge R'(w',u'))} (c \wedge V'(u',\psi)) && (3.29) \\
&= t \wedge \bigvee_{u' \in W'_t} \left[\Big(\bigvee_{c \in D_\tau(t \wedge R'(w',u'))} c \Big) \wedge V'(u',\psi) \right] && (3.30) \\
&\leq t \wedge \bigvee_{u' \in W'_t} [R'(w',u') \wedge V'(u',\psi)] && (3.31) \\
&= t \wedge \bigvee_{u' \in W'} [R'(w',u') \wedge V'(u',\psi)] && (3.32) \\
&= t \wedge V'(w', \Diamond\psi).
\end{aligned}$$

From (3.26) to (3.27) we used the fact that for $u \in W \backslash W_t$, $t \wedge R(w,u) = 0$ and so $t \wedge R(w,u) \wedge V(u,\psi) = 0$, hence such a u does not contribute anything in the preceding join. From (3.28) to (3.29) we use the definition of w'_t and for any pair (u, c) in the (3.28), the forth condition gives a $u' \in W'$ such that $c \leq R'(w', u')$ and $(u, u') \in Z(c)$. By the Inductive Hypothesis, $c \wedge V(u,\psi) = c \wedge V'(u',\psi)$ and $c \leq t \wedge R(w,u)$ implies

$c \leq t \wedge R'(w', u')$. From (3.30) to (3.31) we used the fact that

$$\bigvee_{c \in D_\tau(t \wedge R(w,u))} c \leq t \wedge R(w,u) = t \wedge R'(w', u') \leq R'(w', u').$$

From (3.31) to (3.32) we used the fact that for $u' \in W' \backslash W'_t$, $t \wedge R'(w', u') = 0$ and so $t \wedge R'(w', u') \wedge V'(u', \psi) = 0$, hence such a u' does not contribute anything in the preceding join.
The proof of the other implication is symmetric.

Suppose $\phi = @_{\mathbf{i}}\psi$. Then

$$\begin{aligned} t \wedge V(w, @_{\mathbf{i}}\psi) &= t \wedge V(n(\mathbf{i}), \psi) \\ & t \wedge V'(n'(\mathbf{i}), \psi) && \text{(by IH and } (n(\mathbf{i}), n'(\mathbf{i})) \in Z(t)) \\ &= t \wedge V'(w', @_{\mathbf{i}}\psi). \end{aligned}$$

Therefore the case in which t is join-irreducible is complete. Suppose now that t is not join-irreducible. Then

$$\begin{aligned} t \wedge V(w, \phi) &= \left(\bigvee_{c \in D_\tau(t)} c \right) \wedge V(w, \phi) \\ &= \bigvee_{c \in D_\tau(t)} (c \wedge V(w, \phi)) \\ &= \bigvee_{c \in D_\tau(t)} (c \wedge V'(w', \phi)) && \text{(since } c \text{ is join-irreducible)} \\ &= \left(\bigvee_{c \in D_\tau(t)} c \right) \wedge V'(w', \phi) \\ &= t \wedge V'(w', \phi). \end{aligned}$$

□

The whole section has been building up to the next corner stone theorem which is essentially the converse of the preceding theorem; that is, truth invariance implies weak bisimulation but only for a special set of model, namely the image-finite models. We constrain our models to a certain condition and prove this converse for this certain class of models. The proof does not go through with general models. The definition of the *image-finite models* was given for classical modal logic in [3] and was later extended to the many-valued modal logic in [17, Definition 3.13]. We will also use the definition given in [17].

Definition 3.4.9. A model $\mathfrak{M} = (W, R, n, V)$ is called t-image-finite if for every $w \in W$, the set $W_w^t = \{u \in W : t \wedge R(w, u) \neq 0\}$ is finite.

Lemma 3.4.10. Let τ be a Heyting algebra and let $\mathfrak{M} = (W, R, n, V)$ be a t-image-finite where $t \in \tau (t \neq 0)$. If $t' \leq t$, then $\mathfrak{M}$ is t'-image-finite.

Proof. Since $\mathfrak{M}$ is t-image-finite, the set $W_w^t = \{u \in W : t \wedge R(w,u) \neq 0\}$ is finite. Suppose that $t' \leq t$. Since $t \wedge R(w,u) \neq 0$ for all $u \in W$, we have $t' \wedge t \wedge R(w,u) \neq 0 \wedge t'$ for all $u \in W$. Then we have that $t' \wedge R(w,u) \neq 0$ for all $u \in W$. Since W_w^t is finite, it follows that $W_w^{t'} = \{u \in W : t' \wedge R(w,u) \neq 0\}$ is also finite. Hence $\mathfrak{M}$ is t'-image-finite. □

The following theorem states that for t-image-finite τ-models, t-equivalence implies weak t-bisimilarity. The proof for many-valued modal case is given in [17]. We will show each case in detail and extend the theorem to the many-valued hybrid case.

Theorem 3.4.11. Let $t \in \tau$ $(t \neq 0)$, $\mathfrak{M} = (W, R, n, V)$ and $\mathfrak{M}' = (W', R', n', V')$ be t-image-finite models and $w \in W$ and $w' \in W'$ two states. If $t \wedge V(w, \phi) = t \wedge V'(w', \phi)$ for every formula ϕ, then $\mathfrak{M}, w \leftrightsquigarrow_t \mathfrak{M}', w'$.

Proof. Define the function $Z : \tau \backslash \{0\} \to 2^{W \times W'}$ so that for every $w \in W$, every $w' \in W'$ and every $d \in \tau \backslash \{0\}$, $(w, w') \in Z(d)$ iff $d \wedge V(w, \phi) = d \wedge V'(w', \phi)$. We show that Z is a weak bisimulation.

- (consistency) Suppose that $(w, w') \in Z(t_1 \vee t_2)$. Then for every ϕ, $(t_1 \vee t_2) \wedge V(w, \phi) = (t_1 \vee t_2) \wedge V'(w', \phi)$. Then $t_1 \wedge (t_1 \vee t_2) \wedge V(w, \phi) = t_1 \wedge (t_1 \vee t_2) \wedge V'(w', \phi)$ which implies that $t_1 \wedge V(w, \phi) = t_1 \wedge V'(w', \phi)$ since $t_1 \leq t_1 \vee t_2$. Therefore $(w, w') \in Z(t_1)$. Similarly, taking the meet with t_2 implies that $(w, w') \in Z(t_2)$. Hence $(w, w') \in Z(t_1) \cap Z(t_2)$.

  Conversely, suppose that $(w, w') \in Z(t_1) \cap Z(t_2)$. Then for every ϕ, $t_1 \wedge V(w, \phi) = t_1 \wedge V'(w', \phi)$ and $t_2 \wedge V(w, \phi) = t_2 \wedge V'(w', \phi)$ By distribution, $(t_1 \vee t_2) \wedge V(w, \phi) = (t_1 \vee t_2) \wedge V'(w', \phi)$. Hence $(w, w') \in Z(t_1 \vee t_2)$.

- From the definition of Z we have that $d \wedge V(w, p) = d \wedge V'(w', p)$ for every $p \in \Phi$.

- Suppose that $n(\mathbf{i}) = w$ and $n'(\mathbf{i}) = w'$ for $\mathbf{i} \in \Omega$. We want to show that $(w, w') \in Z(d)$. Suppose that, for the sake of a contradiction, $(w, w') \notin Z(d)$. Then we have that $d \wedge V(w, \phi) \neq d \wedge V'(w', \phi)$. Now we consider cases:

  – Case 1: $\phi = \mathbf{i}$:

    Then $d \wedge V(w, \phi) = d \wedge 1 = d$ and $d \wedge V'(w', \phi) = d \wedge 1 = d$ which is a contradiction.

  – Case 2: $\phi \neq \mathbf{i}$:

    Then $d \wedge V(w, \phi) = d \wedge 0 = 0$ and $d \wedge V'(w', \phi) = d \wedge 0 = 0$ which is a contradiction.

- (forth condition) Suppose for the sake of a contradiction that Z does not satisfy the forth condition. Then there exist a join-irreducible value $d \in J(\tau)$, a pair $(w, w') \in Z(d)$, a state $u \in W$ such that $d \wedge R(w,u) \neq 0$ and a join-irreducible value $c \in D_\tau(d \wedge R(w,u))$ such that for every $u' \in W'$, if $c \leq R'(w',u')$ then (u,u') does not belong to $Z(c)$.

  Since $\mathfrak{M}'$ is t-image-finite, the set $W'_c = \{u' \in W' : c \leq R'(w',u')\}$ is finite. We first show that W'_c is non-empty. We have

$$\begin{aligned} c &\leq d \wedge R(w,u) \\ &\leq d \wedge \bigvee_{u \in W} [R(w,u) \wedge 1] \\ &= d \wedge V(w, \Diamond 1) \\ &= d \wedge V'(w', \Diamond 1) \\ &= d \wedge \bigvee_{u' \in W'} \left[R'(w',u') \wedge 1\right] \\ &= \bigvee_{u' \in W'} \left[d \wedge R'(w',u')\right]. \end{aligned}$$

  From Lemma 2.1.25 (7) there exists some $u' \in W'$ such that

$$c \leq d \wedge R'(w',u') \leq R'(w',u').$$

  Therefore W'_c is non-empty. Suppose that $W'_c = \{u'_1, u'_2, \ldots, u'_k\}$. Then for every for i, $1 \leq i \leq k$, (u, u'_i) does not belong to $Z(c)$. Then there exists a formula ϕ_i such that $c \wedge V(u,\phi_i) \neq c \wedge V'(u'_i,\phi_i)$. We will define a new formula ψ_i such that $c \wedge V(u,\psi_i) = c$ and $c \wedge V'(u'_i,\psi_i) < c$. Let $a_i = c \wedge V(u,\phi_i)$ and $b_i = c \wedge V'(u'_i,\phi_i)$. Let $\mathbf{a}_i$ corresponds to $a_i \in \tau$. We consider two cases:

  1. $a_i \leq b_i$. Define $\psi_i = \phi_i \to \mathbf{a}_i$. Then

$$\begin{aligned} c \wedge V(u,\psi_i) &= c \wedge V(u, \phi_i \to \mathbf{a}_i) \\ &= c \wedge (V(u,\phi_i) \Rightarrow V(u,\mathbf{a}_i)) \\ &= c \wedge (V(u,\phi_i) \Rightarrow a_i) \\ &= c \wedge ((c \wedge V(u,\phi_i)) \Rightarrow (c \wedge a_i)) && \text{(by Lemma2.1.25 (7))} \\ &= c \wedge (a_i \Rightarrow a_i) && \text{(since } a_i \leq c). \\ &= c. \end{aligned}$$

     Similarly,

$$\begin{aligned} c \wedge V'(u'_i,\psi_i) &= c \wedge V'(u'_i, \phi_i \to \mathbf{a}_i) \\ &= c \wedge (V'(u'_i,\phi_i) \Rightarrow V'(u'_i,\mathbf{a}_i)) \\ &= c \wedge (V'(u'_i,\phi_i) \Rightarrow a_i) \\ &= c \wedge ((c \wedge V'(u'_i,\phi_i)) \Rightarrow (c \wedge a_i)) && \text{(by Lemma2.1.25 (6))} \\ &= c \wedge (b_i \Rightarrow a_i). \end{aligned}$$

Suppose to the contrary that $c \wedge (b_i \Rightarrow a_i) = c$. Then $c \leq (b_i \Rightarrow a_i)$. Hence $c \wedge b_i \leq a_i$ by (by Lemma2.1.25 (5)). Note that $b_i = c \wedge V'(u_i', \phi_i)$. So we have that $c \wedge b_i = c \wedge c \wedge V'(u_i', \phi_i)$. From the idempotency, we have that $c \wedge b_i = c \wedge V'(u_i', \phi_i)$. Hence, $c \wedge b_i = b_i$.

Therefore $c \wedge b_i = b_i \leq a_i$. Hence $a_i = b_i$, which is a contradiction. Hence $c \wedge V'(u_i', \psi_i) < c$.

2. $a_i \not\leq b_i$. Define $\psi_i = \mathbf{a}_i \to \phi_i$. Then

$$\begin{aligned} c \wedge V(u, \psi_i) &= c \wedge V(u, \mathbf{a}_i \to \phi_i) \\ &= c \wedge (V(u, \mathbf{a}_i) \Rightarrow V(u, \phi_i)) \\ &= c \wedge (a_i \Rightarrow V(u, \phi_i)) \\ &= c \wedge ((c \wedge a_i) \Rightarrow (c \wedge V(u, \phi_i))) \\ &= c \wedge (a_i \Rightarrow a_i) \\ &= c. \end{aligned}$$

Similarly,

$$\begin{aligned} c \wedge V'(u_i', \psi_i) &= c \wedge V'(u_i', \mathbf{a}_i \to \phi_i) \\ &= c \wedge (V'(u_i', \mathbf{a}_i) \Rightarrow V'(u_i', \phi_i)) \\ &= c \wedge (a_i \Rightarrow V'(u_i', \phi_i)) \\ &= c \wedge ((c \wedge a_i) \Rightarrow (c \wedge V'(u_i', \phi_i))) \\ &= c \wedge (a_i \Rightarrow b_i). \end{aligned}$$

Suppose to the contrary that $c = c \wedge (a_i \Rightarrow b_i)$. Then, $c \leq (a_i \Rightarrow b_i)$. Hence $c \wedge a_i \leq b_i$ by Lemma(2.1.25 (5)). Since $a_i = c \wedge a_i$, it follows that $a_i \leq b_i$ which is a contradiction. Therefore, $c \wedge V'(u_i', \psi_i) < c$.

Let $\psi = \bigwedge_{1 \leq i \leq k} \psi_i$. Then

$$\begin{aligned} c \wedge V(u, \psi) &= c \wedge V\left(u, \bigwedge_{1 \leq i \leq k} \psi_i\right) \\ &= c. && (\text{ since } c \wedge V(u, \psi_i) = c \text{ for all } 1 \leq i \leq k) \end{aligned}$$

Similarly $c \wedge V'(u_i', \psi) = c \wedge V'\left(u_i', \bigwedge_{1 \leq i \leq k} \psi_i\right) < c$. Note that this is also saying that $c \leq V(u, \psi)$ and $c \not\leq V'(u_i', \psi)$ for every i, $1 \leq i \leq k$. Moreover,

$$V(w, \Diamond\psi) = \bigvee_{v \in W} [R(w, v) \wedge V(v, \psi)] \geq R(w, u) \wedge V(u, \psi) \geq c \wedge c = c \tag{3.33}$$

Since $(w, w') \in Z(d)$, we have, in particular, that $d \wedge V(w, \Diamond\psi) = d \wedge V'(w', \Diamond\psi)$ which implies that $c \wedge V(w, \Diamond\psi) = c \wedge V'(w', \Diamond\psi)$ since $c \leq d$. From Equation 3.33 we have that $V(w, \Diamond\psi) \geq c$. So it follows from the idempotency that $c \wedge V(w, \Diamond\psi) \geq c$. Hence, $c \wedge V'(w', \Diamond\psi) \geq c$, which implies that $V'(w', \Diamond\psi) \geq c$. Therefore, $c \wedge V'(w', \Diamond\psi) = c$.

Hence,

$$
\begin{aligned}
c &\leq V'(w', \Diamond\psi) \\
&= \bigvee_{u' \in W'} \left[R'(w', u') \wedge V'(u', \psi)\right] \\
&= \bigvee_{u' \in W'_c} \left[R'(w', u') \wedge V'(u', \psi)\right] \vee \bigvee_{u' \in W' \setminus W'_c} \left[R'(w', u') \wedge V'(u', \psi)\right] \\
&\leq \bigvee_{u' \in W'_c} \left[V'(u', \psi)\right] \vee \bigvee_{u' \in W' \setminus W'_c} \left[R'(w', u')\right].
\end{aligned}
$$

Since c is a join-irreducible element, Lemma(2.1.25 (5)) implies that either there exists an $u' \in W'_c$ such that $c \leq V'(u', \psi)$ which contradicts our assumption or there exists an $u' \in W' \setminus W'_c$ such that $c \leq R'(w', u')$ which contradicts equation 3.33.

The back condition is proved similar to the forth condition, in the opposite direction.

□

Chapter 4

Correspondence for Many-Valued Hybrid Logic

In this chapter we will pursue one of the central aims of this dissertation. We will establish the *correspondence theory* between many-valued first-order logic and many-valued hybrid logic. Hence we will investigate the connections between first-order definable properties of τ-frames and the many-valued hybrid formulas that are valid in them. Our goal in this chapter is to extend the ALBA algorithm to many-valued hybrid setting. The correspondence theory between many-valued first-order logic and many-valued modal logic was established in [7]. Our purpose is to further extend ALBA to many-valued hybrid logic and therefore establish the correspondence theory between many-valued first-order logic and many-valued hybrid logic. The ALBA algorithm was shown to succeed on a class of formulas that is strictly larger than the *Sahlqvist class*, namely the class of *inductive formulas*. This is a strictly *sufficient condition* for hybrid formulas to have a first-order local frame correspondent. The success of the algorithm on the inductive class of formulas was proved for the 2-valued hybrid logic [13] and it was also proved for the many-valued modal logic [7]. We will combine these two results to formulate the extension of ALBA algorithm. The extension of the algorithm that we will formulate will be referred to as the MV-Hybrid ALBA. We will also prove the correctness of the MV-Hybrid ALBA to establish the correspondence theory between many-valued hybrid logic and many-valued first-order logic.

We will first introduce many-valued first-order logic (and an extension of it) which will be the correspondence languages for many-valued hybrid language (and an extension of it).

4.1 The Basic Many-Valued First-Order Language and its Extended Language

We now introduce the *basic many-valued first-order language* L_τ^{FO}, and later its extension L_τ^{FO+} which will serve as *correspondence* languages for L_τ and L_τ^+, respectively. A perfect Heyting algebra τ will serve as a truth-value space for the languages L_τ^{FO} and L_τ^{FO+}. We will also define a standard translation between L_τ and L_τ^+ and their appropriate correspondence languages and it will be proven that t-truth and t-validity are *invariant* under the standard translation.

4.1.1 The Basic Many-Valued First-Order Language

The basic many-valued first-order language L_τ^{FO} contains the following:

1. Logical Symbols:
   (a) A set of individual variables VAR, the elements of which will be denoted by $x_1, x_2, x_3, \ldots$
   (b) The connectives $\wedge, \vee$ and $\rightarrow$
   (c) The quantifiers $\forall$ and $\exists$
   (d) Equality =
   (e) Truth constants $\mathbf{t}$ for each $t \in \tau$
2. Nonlogical Symbols:
   (a) Unary predicate symbols $P_1, P_2, P_3 \ldots$
   (b) Constant symbols $\mathbf{d_i}$ for each $\mathbf{i} \in \Omega$
   (c) Binary relation R

Terms of L_τ^{FO} are elements of VAR together with the constant symbols, and the set of terms is denoted by $TERM_\tau$. The set of formulas of L_τ^{FO} is denoted by $FORM_\tau$ and the formulas are inductively defined as

1. If $s_1, s_2 \in TERM_\tau$, then $(s_1 = s_2)$, $R(s_1, s_2)$, $P(s_1) \in FORM_\tau$
2. If $t \in \tau$, then $\mathbf{t} \in FORM_\tau$
3. If $A, B \in FORM_\tau$, then $A \wedge B, A \vee B, A \rightarrow B \in FORM_\tau$
4. If $A \in FORM_\tau$ and $x_n \in VAR$, then $\forall x_n A, \exists x_n A \in FORM_\tau$
5. If $\mathbf{d_i}$ is a constant and $A \in FORM_\tau$, then $\forall \mathbf{d_i} A \in FORM_\tau$ and $\exists \mathbf{d_i} A \in FORM_\tau$
6. Every formula of L_τ^{FO} can be obtained using a finite number of applications of item 1 through 5.

The connective $\neg$ is defined as $\neg A = A \to 0$. We now define the notion of *interpretations* on L_τ^{FO}. From a correspondence theoretic point of view, interpretations are the same mathematical objects as models.

Definition 4.1.1. An *interpretation* I of L_τ^{FO} is a structure consisting of the following

1. A non-empty set W^I, called the domain of I
2. For each unary predicate symbol P_n of L_τ^{FO}, a τ-valued unary predicate $P_n^I : W \to \tau$
3. For each constant symbol $\mathbf{d_i}$, an element d_i^I of W^I
4. For the binary relation symbol R, a binary τ-valued relation $R^I : (W \times W) \to \tau$

Definition 4.1.2. Let I be an interpretation for L_τ^{FO}. An *assignment* on I is a function v such that $v : VAR \to W^I$

Definition 4.1.3. Two assignments v and v' on an interpretation I are *x_n-variants* (notation : $v' \sim_{x_n} v$) if $v(x_k) = v'(x_k)$ for all $k \neq n$.

Definition 4.1.4. Given an interpretation I, an assignment v can be extended to formulas of L_τ^{FO} such that $v : FORM_\tau \to \tau$. This assignment interprets formulas of L_τ^{FO} as follows

1. $v(s_1 = s_2) = \begin{cases} 1 & \text{if } v(s_1) = v(s_2) \\ 0 & \text{otherwise} \end{cases}$
2. $v\left(R\left(s_1, s_2\right)\right) = R^I\left(v\left(s_1\right), v\left(s_2\right)\right)$
3. $v\left(P_n\left(s_1\right)\right) = P_n^I\left(v\left(s_1\right)\right)$
4. $v(\mathbf{t}) = t$
5. $v(\alpha \wedge \beta) = v(\alpha) \wedge^\tau v(\beta)$
6. $v(\alpha \vee \beta) = v(\alpha) \vee^\tau v(\beta)$
7. $v(\alpha \to \beta) = v(\alpha) \Rightarrow v(\beta)$
8. $v(\forall x_n \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{x_n} v\}$
9. $v(\exists x_n \alpha) = \bigvee\{v'(\alpha) \mid v' \sim_{x_n} v\}$

For the defined $\neg$, we will interpret $\neg A$ as $v(\neg A) = v(A) \Rightarrow 0$.

4.1.2 The Extended Many-Valued First-Order Language

Since the extended many-valued hybrid language L_τ^+ of L_τ has additional modal operators $\blacklozenge$ and $\blacksquare$, the operators $@^\#$ and $@^\flat$, J-variables and M-variables, additional symbols must be added to the language L_τ^{FO+} in order to define a suitable correspondence language for L_τ^+.

The extended many-valued first-order language L_τ^{FO+} contains the following and all the nonlogical symbols of L_τ^{FO}:

1. Nonlogical Symbols:
   (a) Constant symbols $\mathbf{c_j}$ and $\mathbf{c_m}$ for each $\mathbf{j} \in J$-VAR and for each $\mathbf{m} \in M$-VAR.
   (b) Truth-constant symbols $\mathbf{C_j}$ and $\mathbf{C_m}$ for each $\mathbf{j} \in J$-VAR and for each $\mathbf{m} \in M$-VAR.

Note that the language L_τ^{FO} and L_τ^{FO+} have exactly the same logical symbols, so we omitted that in the definition. Terms of L_τ^{FO+} will be elements of $TERM_\tau$, together with the constant symbols $\mathbf{c_j}$ and $\mathbf{c_m}$. The new set of terms will be denoted by $TERM_\tau^+$. The set of formulas of L_τ^{FO+} is denoted by $FORM_\tau^+$ and the formulas are inductively defined as

1. If $\alpha \in FORM_\tau$, then $\alpha \in FORM_\tau^+$.
2. If $\mathbf{c_j}$ and $\mathbf{c_m}$ are constants and $x_n \in VAR$, then $(\mathbf{c_j} = x_n), (\mathbf{c_m} = x_n) \in FORM_\tau^+$.
3. If $\mathbf{C_j}$ and $\mathbf{C_m}$ are truth constants, then $\mathbf{C_j}, \mathbf{C_m} \in FORM_\tau^+$.
4. If $A, B \in FORM_\tau^+$, then $A \wedge B, A \vee B, A \to B \in FORM_\tau^+$.
5. If $A \in FORM_\tau^+$ and $x_n \in VAR$, then $\forall x_n A, \exists x_n A \in FORM_\tau^+$
6. If $\mathbf{d_i}$ is a constant and $A \in FORM_\tau^+$, then $\forall \mathbf{d_i} A \in FORM_\tau^+$ and $\exists \mathbf{d_i} A \in FORM_\tau^+$
7. Every formula of L_τ^{FO+} can be obtained using a finite number of applications of item 1 through 6.

Definition 4.1.5. An *interpretation* I of L_τ^{FO+} is a structure consisting of the following

1. A non-empty set W^I, called the domain of I.
2. For each unary predicate symbol P_n of L_τ^{FO+}, a many-valued unary predicate $P_n^I : W^I \to \tau$.
3. For the binary relation symbol R, a binary function R^I such that $R^I : (W^I \times W^I) \to \tau$.

4. For the constants symbols $\mathbf{c_j}$ and $\mathbf{c_m}$ of L_τ^{FO+}, an element c_j^I and c_m^I of W^I, respectively.

5. For the truth constants symbols $\mathbf{C_j}$ and $\mathbf{C_m}$ of L_τ^{FO+}, an element C_j^I and C_m^I in τ, respectively.

We now define the notion of an assignment on an interpretation I for the language L_τ^{FO+}.

Definition 4.1.6. Let I be an interpretation for L_τ^{FO+}. An *assignment* on I is a function v such that $v : VAR \to W^I$.

Definition 4.1.7. Two assignments v and v' on an interpretation I are δ_n*-variants* (notation: $v' \sim_{\delta_n} v$) if $v(\delta_k) = v'(\delta_k)$ for all $k \neq n$ and δ is a variable or a constant indexed by k.

An assignment can be extended to formulas of L_τ^{FO+}. The extended assignment will interpret the formulas of L_τ^{FO+} as in the assignment on L_τ^{FO} with the additional clauses for L_τ^{FO+}.

Definition 4.1.8. Given an interpretation I, an assignment v can be extended to formulas of L_τ^{FO+} such that $v : FORM_\tau^+ \to \tau$. This assignment interprets formulas of L_τ^{FO+} as follows

1. $v(\mathbf{C_j}) = C_j^I$

2. $v(\mathbf{C_m}) = C_m^I$

4.1.3 The Basic Many-Valued Second-Order Language and its Extension

In the 2-valued setting, there are notions that exceed the expressive power of first-order logic, however, these notions were expressible in second-order logic. For example:

$$\text{Well-Orderedness: } \forall \mathbf{P}\,(\exists x \mathbf{P}(x) \to \exists y\,(\mathbf{P}(y) \wedge \neg\exists z(\mathbf{P}(z) \wedge (z \leq y))))$$

can be expressed in second-order logic, but not in first-order logic. In correspondence theory, hybrid formulas always correspond to a second-order formula, which can sometimes be broken down to a first-order formula (but this is not always the case).

The language L_τ^{SO} will be much like L_τ^{FO}, but it will be also be able to quantify over the unary predicate symbols P_n of L_τ^{FO}. Hence, $\forall P_n \alpha$ and $\exists P_n \alpha$ are also formulas of L_τ^{SO}.

Any assignment on L_τ^{SO} should still interpret first-order formulas as in L_τ^{FO}, however, the definition of an assignment must vary to allow for quantification over second-order variables. An assignment on L_τ^{SO} is defined as follows:

Definition 4.1.9. Given an interpretation I for L_τ^{SO}, an assignment v on I is a mapping $v : (VAR \cup \{P_n\} \cup \{\mathbf{d_i}\} \cup \{\mathbf{c_j}\} \cup \{\mathbf{c_m}\} \to (W \cup \{f : W \to \tau\})$ such that

1. $v(x_n) = w$ for some w in W, where $x_n \in VAR$.
2. $v(P_n) = f$, for some $f : W \to \tau$, where P_n is a predicate symbol.
3. $v(\mathbf{d_i}) = f$, for some $f : W \to \tau$, where $f(w) = 1$ for exactly on w in W and $f(w_0) = 0$ for all $w_0 \neq w$ and $\mathbf{i} \in \Omega$.
4. $v(\mathbf{c_j}) = f$, for some $f : W \to \tau$, where $f(w) \in J^\infty(\tau)$ for exactly one w in W and $f(w_0) = 0$ for all $w_0 \neq w$ and $\mathbf{j} \in J$-VAR.
5. $v(\mathbf{c_m}) = f$, for some $f : W \to \tau$, where $f(w) \in M^\infty(\tau)$ for exactly one w in W and $f(w_0) = 1$ for all $w_0 \neq w$ and $\mathbf{m} \in M$-VAR.

We have used the same notation v for an assignment in L_τ^{FO} and L_τ^{SO}. When used, it will be clear from the context which we referring to.

Definition 4.1.10. 1. Two assignments v and v' are P_n-variants (notation: $v' \sim_{P_n} v$) if $v(P_k) = v'(P_k)$ for all $k \neq n$.

2. Two assignments v and v' on an interpretation I are $\mathbf{c_j}$ (notation: $v' \sim_{\mathbf{c_j}} v$) if $v(\mathbf{c_k}) = v'(\mathbf{c_k})$ for all $\mathbf{k} \neq \mathbf{j}$.
3. Two assignments v and v' on an interpretation I are $\mathbf{c_m}$ (notation: $v' \sim_{\mathbf{c_m}} v$) if $v(\mathbf{c_n}) = v'(\mathbf{c_n})$ for all $\mathbf{n} \neq \mathbf{m}$.
4. Two assignments v and v' on an interpretation I are $\mathbf{d_i}$ (notation: $v' \sim_{\mathbf{d_i}} v$) if $v(\mathbf{d_i'}) = v'(\mathbf{d_{i'}})$ for all $\mathbf{i}' \neq \mathbf{i}$.

A second-order assignment v will interpret a first-order formula as in L_τ^{FO}. Second-order formulas will be interpreted as follows:

Definition 4.1.11. Let $s \in TERM_\tau$, then we have the following

1. $v(P_n(s)) = v(P_n)(v(s))$.
2. $v(\forall P_n \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{P_n} v\}$.
3. $v(\exists P_n \alpha) = \bigvee\{v'(\alpha) \mid v' \sim_{P_n} v\}$.
4. $v(\forall \mathbf{c_j} \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{\mathbf{c_j}} v\}$.
5. $v(\forall \mathbf{c_m} \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{\mathbf{c_m}} v\}$.

6. $v(\forall \mathbf{d_i} \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{\mathbf{d_i}} v\}$.

7. $v(\exists \mathbf{c_j} \alpha) = \bigvee\{v'(\alpha) \mid v' \sim_{\mathbf{c_j}} v\}$.

8. $v(\exists \mathbf{c_m} \alpha) = \bigvee\{v'(\alpha) \mid v' \sim_{\mathbf{c_m}} v\}$.

9. $v(\exists \mathbf{d_i} \alpha) = \bigvee\{v'(\alpha) \mid v' \sim_{\mathbf{d_i}} v\}$.

Extended Many-Valued Second-Order Language The truth constant symbols $\mathbf{C_j}$ and $\mathbf{C_m}$ introduced in L_τ^{FO+} can be viewed as nullary many-valued predicates. The extended many-valued second-order language L_τ^{SO+}, which is intended to be the correspondence language for L_τ^+, will allow for quantification over truth constants symbols. These shall be seen as quantification over τ.

The language L_τ^{SO+} contains additional terms $\mathbf{c_j}$, $\mathbf{c_m}$ and $\mathbf{d_i}$ and additional formulas containing $\mathbf{C_j}$ and $\mathbf{C_m}$.

An assignment on the language L_τ^{SO+} is defined as follows:

Definition 4.1.12. Given an interpretation I for L_τ^{SO}, an assignment v on I is a mapping $v : (VAR \cup \{P_n\} \cup \{\mathbf{d_i}\} \cup \{\mathbf{c_j}\} \cup \{\mathbf{c_m}\} \cup \{\mathbf{C_j}\} \cup \{\mathbf{C_m}\} \to (W \cup \{f : W \to \tau\})$ such that

1. $v(x_n) = w$ for some w in W, where $x_n \in VAR$.

2. $v(P_n) = f$, for some $f : W \to \tau$, where P_n is a predicate symbol.

3. $v(\mathbf{d_i}) = f$, for some $f : W \to \tau$, where $f(w) = 1$ for exactly on w in W and $f(w_0) = 0$ for all $w_0 \neq w$ and $\mathbf{i} \in \Omega$.

4. $v(\mathbf{c_j}) = f$, for some $f : W \to \tau$, where $f(w) \in J^\infty(\tau)$ for exactly one w in W and $f(w_0) = 0$ for all $w_0 \neq w$ and $\mathbf{j} \in J$-VAR.

5. $v(\mathbf{c_m}) = f$, for some $f : W \to \tau$, where $f(w) \in M^\infty(\tau)$ for exactly one w in W and $f(w_0) = 1$ for all $w_0 \neq w$ and $\mathbf{m} \in M$-VAR.

Definition 4.1.13. 1. Two assignments v and v' on I are $\mathbf{C_j}$-variant (notation: $v' \sim_{\mathbf{C_j}} v$) if $v(\mathbf{C_k}) = v'(\mathbf{C_k})$ for all $\mathbf{k} \neq \mathbf{j}$.

2. Two assignments v and v' on I are $\mathbf{C_m}$-variant (notation: $v' \sim_{\mathbf{C_m}}$ v) if $v(\mathbf{C_n}) = v'(\mathbf{C_n})$ for all $\mathbf{n} \neq \mathbf{m}$.

The second-order assignment v will interpret $\mathbf{c_j}$, $\mathbf{c_m}$, $\mathbf{d_i}$ and formulas containing $\mathbf{C_j}$ and $\mathbf{C_m}$ as in the language L_τ^{SO} case. Quantification over $\mathbf{C_j}$ and $\mathbf{C_m}$ will be handled as follows:

Definition 4.1.14. 1. $v(\forall \mathbf{C_j} \alpha) = \bigwedge\{v'(\alpha) \mid v' \sim_{\mathbf{C_j}} v\}$.

2. $v(\forall \mathbf{C_m} \alpha) = \bigwedge \{v'(\alpha) \mid v' \sim_{\mathbf{C_m}} v\}$.

3. $v(\exists \mathbf{C_j} \alpha) = \bigvee \{v'(\alpha) \mid v' \sim_{\mathbf{C_j}} v\}$.

4. $v(\exists \mathbf{C_m} \alpha) = \bigvee \{v'(\alpha) \mid v' \sim_{\mathbf{C_m}} v\}$.

4.1.4 Standard Translation

As in 2-valued correspondence theory, the *standard translation* is the *link* between the language $L_\tau(L_\tau^+)$ and language $L_\tau^{FO}(L_\tau^{FO+})$. The standard translation "translates" a many-valued hybrid formula into a many-valued first-order formula in such a way that a-truth and a-validity, where $a \in \tau$, of the formula is preserved in the corresponding language. These are also preserved when we consider the *second-order* standard translation. Such results are very crucial in our research as it plays a big role in proving one of the main theorems of the dissertation, the correctness of Hybrid MV-ALBA (Theorem 4.2.5).

Definition 4.1.15. Given a τ-frame $\mathfrak{F} = (W, R)$ for L_τ^+ and a valuation V on $\mathfrak{F}$, the corresponding first-order interpretation of L_τ^{FO}, denoted by $\mathfrak{M}$, consists of:

1. $W^I = W$.

2. $R^I = R$.

3. $P_n^I(w) = V(w, p_n)$ for $w \in W$ and $p_n \in \Phi$.

4. $d_i^I = n(\mathbf{i})$ for each $\mathbf{i} \in \Omega$.

5. $c_j^I = w_0$ where $V(w_0, \mathbf{j}) \neq 0$ for each $\mathbf{j} \in J\text{-}VAR$.

6. $C_j^I = V(w_o, \mathbf{j}) = J \in J^\infty(\tau)$ for each $\mathbf{j} \in J\text{-}VAR$.

7. $c_m^I = w_1$ where $V(w_1, \mathbf{m}) \neq 1$ for each $\mathbf{m} \in M\text{-}VAR$.

8. $C_m^I = V(w_1, \mathbf{m}) = M \in M^\infty(\tau)$ for each $\mathbf{m} \in M\text{-}VAR$.

With regards to definition (4.1.15), item 4, since nominals are only true at a unique state in a model, we have that $V(w_2, \mathbf{i}) = 1$ which follows that $w_2 = n(\mathbf{i})$.

Definition 4.1.16. Let x be a first-order individual variable. The *standard translation* ST_x taking hybrid formulas of L_τ(or L_τ^+) to formulas of L_τ^{FO}(or L_τ^{FO+}) is defined by the following:

1. $ST_x(p_n) = P_n(x)$.

2. $ST_x(\mathbf{t}) = \mathbf{t}$.

3. $ST_x(\phi \vee \psi) = ST_x(\phi) \vee ST_x(\psi)$.

4. $ST_x(\phi \wedge \psi) = ST_x(\phi) \wedge ST_x(\psi)$.
5. $ST_x(\phi \to \psi) = ST_x(\phi) \to ST_x(\psi)$.
6. $ST_x(\Diamond\psi) = \exists y \left(Rxy \wedge ST_y(\psi)\right)$.
7. $ST_x(\Box\psi) = \forall y \left(Rxy \to ST_y(\psi)\right)$.
8. $ST_x(\blacklozenge\psi) = \exists y \left(Ryx \wedge ST_y(\psi)\right)$.
9. $ST_x(\blacksquare\psi) = \forall y \left(Ryx \to ST_y(\psi)\right)$.
10. $ST_x(\mathbf{j}) = (x = \mathbf{c_j}) \wedge \mathbf{C_j}$.
11. $ST_x(\mathbf{m}) = (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$.
12. $ST_x(\mathbf{i}) = (x = \mathbf{d_i})$
13. $ST_x(@_\mathbf{i}\psi) = \exists y \left(y = \mathbf{d_i} \wedge ST_y(\psi)\right) \equiv \forall y \left(y = \mathbf{d_i} \to ST_y(\psi)\right)$.
14. $ST_x(@^\#_\mathbf{i}\psi) = (x = \mathbf{d_i}) \to \forall y \left(ST_y(\psi)\right)$.
15. $ST_x(@^\flat_\mathbf{i}\psi) = (x = \mathbf{d_i}) \wedge \exists y \left(ST_y(\psi)\right)$.

where y is a variable that has not yet been used in the translation, $p_n \in \Phi$, $\mathbf{i} \in \Omega$, $\mathbf{j} \in J\text{-}VAR$, $\mathbf{m} \in M\text{-}VAR$ and $\mathbf{t}$ is the truth-value constant corresponding to $t \in \tau$.

The following proposition asserts that the truth of a hybrid formula is preserved when the formula is translated to a first-order formula. This is an important results since it allows us to move from language to language while preserving the truth of formulas, and such a result also play a part in proving Theorem 4.2.5. An analogous proposition for many-valued modal logic is found in [7, Proposition 3.29], and its proof is given in detail. We will state the proposition and complete the proof by only considering the clauses for the hybrid setting.

Proposition 4.1.17. Let $\mathfrak{M} = (W, R, n, V)$ be a τ-model, $w \in W$ and ϕ a formula of L^+_τ. Let v be any assignment on the first-order interpretation $\mathfrak{M}$ such that $v(x) = w$. Then:

$$V(w, \phi) = v\left(ST_x(\phi)\right)$$

where x is a variable of L^+_τ.

Proof. The proof is by induction on ϕ.

1. Suppose that $\phi = \mathbf{i}$. There are two cases to consider:
   - Case 1: $V(w, \mathbf{i}) = 0$
     So we have that $d^I_i \neq w$. Let v be an arbitrary assignment on the first-order interpretation $\mathfrak{M}$ such that $v(x) = w$. We have that $v(ST_x(\mathbf{i})) = v(x = \mathbf{d_i})$. But $v(x) = w \neq d^I_i$. So we have that $v(x = \mathbf{d_i}) = 0$ so that $v(ST_x(\mathbf{i})) = 0$ as desired.

- Case 2: $V(w, \mathbf{i}) = 1$
  So we have that $d_i^I = w$. Let v be an arbitrary assignment on the first-order interpretation $\mathfrak{M}$ such that $v(x) = w$. So $v(\mathbf{d_i}) = d_i^I = v(x)$. Hence $v(\mathbf{d_i} = x) = 1$ so that $v(ST_x(\mathbf{i})) = 1$ as desired.

Let ψ be a formula containing fewer connectives than a formula ϕ. Assume the following: If v is an assignment such that $v(y) = u$, then $V(u, \psi) = v(ST_y(\psi))$. Let v be an assignment on the first-order interpretation $\mathfrak{M}$ such that $v(x) = w$.

1. Suppose that $\phi = @_\mathbf{i}\psi$. Then:

$$\begin{aligned} v(ST_x(@_\mathbf{i}\psi)) &= v(\exists y(y = \mathbf{d_i} \wedge ST_y(\psi))) \\ &= \bigvee\{v'(y = \mathbf{d_i} \wedge ST_y(\psi)) \mid v' \sim_y v\} \\ &= \bigvee\{v'(y = \mathbf{d_i}) \wedge v'(ST_y(\psi)) \mid v' \sim_y v\} \\ &= \bigvee\{v'(ST_y(\psi)) \mid v' \sim_y v \text{ and } v'(y) = d_i^I\}. \end{aligned}$$

   Since $x \neq y$ by definition of the standard translation, we have that $v'(x) = v(x) = w$ for any y-variant v' of v. Consider any y-variant v' of v such that $v'(y) = n(\mathbf{i})$. Hence it follows from the inductive hypothesis that $v'(ST_y(\psi)) = V(n(\mathbf{i}), \psi)$. Hence:

$$\begin{aligned} v(ST_x(@_\psi)) &= \bigvee\{v'(ST_y(\psi)) \mid v' \sim_y v \text{ and } v'(y) = d_i^I\} \\ &= \bigvee_{y \in W} (V(n(\mathbf{i}), \psi) \wedge 1) \\ &= \bigvee_{y \in W} (V(n(\mathbf{i}), \psi)) \\ &= V(n(\mathbf{i}), \psi). \end{aligned}$$

2. Suppose that $\phi = @_\mathbf{i}^\# \psi$. Now:

$$\begin{aligned} v(ST_x(@_\mathbf{i}^\#\psi)) &= v((x = \mathbf{d_i}) \to \forall y(ST_y(\psi)) \\ &= v((x = \mathbf{d_i})) \to v(\forall y(ST_y(\psi))) \\ &= v((x = \mathbf{d_i})) \to \bigwedge\{v'(ST_y(\psi)) \mid v' \sim_y v\} \\ &= v((x = \mathbf{d_i})) \to \bigwedge\{V(u, \psi) \mid u \in W\}. \end{aligned}$$

   - Case 1: $w = n(\mathbf{i})$

     Then by definition, we have that $v(\mathbf{d_i}) = n(\mathbf{i})$ which implies that $v(\mathbf{d_i} = n(\mathbf{i}))$=1. Hence,

$$\begin{aligned} v(ST_x(@_\mathbf{i}^\#\psi)) &= 1 \to \bigwedge_{u \in W} V(u, \psi) \\ &= \bigwedge_{u \in W} V(u, \psi). \end{aligned}$$

- Case 2: $w \neq n(\mathbf{i})$

  Then by definition, we have that $v(\mathbf{d_i}) \neq n(\mathbf{i})$ which implies that $v(\mathbf{d_i} = n(\mathbf{i})) = 0$. Hence,

$$\begin{aligned} v(ST_x(@_{\mathbf{i}}^{\#}\psi)) &= 0 \rightarrow \bigwedge_{u \in W} V(u, \psi) \\ &= 1. \end{aligned}$$

  So by the two cases, it follows that $v(ST_x(@_{\mathbf{i}}^{\#}\psi)) = V(w, @_{\mathbf{i}}^{\#}\psi)$.

3. Suppose that $\phi = @_{\mathbf{i}}^{\flat}\psi$. Now:

$$\begin{aligned} v(ST_x(@_{\mathbf{i}}^{\flat}\psi)) &= v((x = \mathbf{d_i}) \wedge \exists y(ST_y(\psi)) \\ &= v((x = \mathbf{d_i})) \wedge v(\exists y(ST_y(\psi))) \\ &= v((x = \mathbf{d_i})) \wedge \bigvee\{v'(ST_y(\psi)) \mid v' \sim_y v\} \\ &= v((x = \mathbf{d_i})) \wedge \bigvee\{V(u, \psi) \mid u \in W\}. \end{aligned}$$

- Case 1: $w = n(\mathbf{i})$

  Then by definition, we have that $v(\mathbf{d_i}) = n(\mathbf{i})$ which implies that $v(\mathbf{d_i} = n(\mathbf{i})) = 1$. Hence,

$$\begin{aligned} v(ST_x(@_{\mathbf{i}}^{\flat}\psi)) &= 1 \wedge \bigvee_{u \in W} V(u, \psi) \\ &= \bigvee_{u \in W} V(u, \psi). \end{aligned}$$

- Case 2: $w \neq n(\mathbf{i})$

  Then by definition, we have that $v(\mathbf{d_i}) \neq n(\mathbf{i})$ which implies that $v(\mathbf{d_i} = n(\mathbf{i})) = 0$. Hence,

$$\begin{aligned} v(ST_x(@_{\mathbf{i}}^{\flat}\psi)) &= 0 \wedge \bigvee_{u \in W} V(u, \psi) \\ &= 0. \end{aligned}$$

  So by the two cases, it follows that $v(ST_x(@_{\mathbf{i}}^{\flat}\psi)) = V(w, @_{\mathbf{i}}^{\flat}\psi)$.

□

Corollary 4.1.18. Let $\mathfrak{M} = (W, R, n, v)$ be a τ-model, $a \in \tau$ and ϕ a formula of L_τ^+. Let x be a variable of L_τ^{FO+}. Then

$$\mathfrak{M}, w \Vdash_a \phi \text{ iff } \mathfrak{M} \vDash_a ST_x(\phi)[x = w]$$

We can make Proposition 4.1.17 more general by substituting the first-order assignment v for a second-order assignment.

Corollary 4.1.19. Let $\mathfrak{M} = (W, R, n, V)$ be a τ-model, $a \in \tau$, $w \in W$ and ϕ a formula of L_τ^+. Let x be a variable of L_τ^{FO+}. Then

$$V(w, \phi) = v(ST_x(\phi))$$

where v is a second-order assignment on the interpretation $\mathfrak{M}$ such that $v(x) = w$ and $v(P) = f$, where $f \in \tau^W$ such that $f(w) = V(w, p)$.

We will skip this proof entirely. The proof is this Corollary is by induction on ϕ. The base case for propositional variable is found in [7]. The other proofs of the other cases are entirely similar to that of Propositional 4.1.17 and mostly uses the fact that a second-order assignment evaluates any first-order formula as a first-order assignment would.

In Propositional 4.1.17 we proved that a hybrid formula and its standard translation have the same truth-value at a state in a τ-model. Now we will use this result to show that the truth-value of a hybrid formula at the state in τ-frame is the same truth-value its standard translation have in the corresponding first-order interpretation.

The proof of the following analogous result is found in [7]. The fact that we are in the hybrid setting does not change the proof. We chose to include this proof in this paper (instead of referencing it) because this result plays a very crucial role in the proof of the *correctness of ALBA algorithm*.

Proposition 4.1.20. Let $\mathfrak{F} = (W, R)$ be a τ-frame, $a \in \tau$, $w \in W$ and ϕ a formula of L_τ^+. Let x be a variable of L_τ^{FO+}. Then

$$\mathfrak{F}, w \Vdash_a \phi \text{ iff } \mathfrak{F} \Vdash_a \forall \mathbf{P} \forall \mathbf{c_j} \forall \mathbf{c_m} \forall \mathbf{C_j} \forall \mathbf{C_m} (ST_x(\phi))[x = w]$$

where $\mathbf{P}$, $\mathbf{c_j}$, $\mathbf{c_m}$, $\mathbf{C_j}$ and $\mathbf{C_m}$ are vectors of many-valued predicate symbols, constant symbols $\mathbf{c_j}$ for each $\mathbf{j} \in J$-VAR, constant symbols $\mathbf{c_m}$ for each $\mathbf{m} \in M$-VAR, truth constant symbols $\mathbf{C_j}$ for each $\mathbf{j} \in J$-VAR and truth constant symbol $\mathbf{C_m}$ for each $\mathbf{m} \in M$-VAR, respectively, occurring in $ST_x(\phi)$.

Proof. We have that

$$\mathfrak{F}, w \Vdash_a \phi \text{ iff } V(w, \phi) \geq a \text{ for all valuation } V \text{ on } \mathfrak{F}$$
$$\text{iif } v(ST_x(\phi)) \geq a \text{ for all assignments } v \text{ on } \mathfrak{F} \text{ such that } v(x) = w$$

by Proposition 4.1.17

$$\text{iff } \mathfrak{F} \Vdash_a \forall \mathbf{P} \forall \mathbf{c_j} \forall \mathbf{c_m} \forall \mathbf{C_j} \forall \mathbf{C_m} (ST_x(\phi))[x = w]$$

The last step follows since $v(ST_x(\phi)) \geq a$ holds for all assignments on $\mathfrak{F}$ such that $v(x) = w$, including all δ-variants for $\delta \in \{\mathbf{P}, \mathbf{c_j}, \mathbf{c_m}, \mathbf{C_j}, \mathbf{C_m}\}$. □

4.2 Correspondence

We know that two formulas ϕ and α are local frame correspondents of each other in the 2-valued if the *validity* of one guarantees the truth of the other in the corresponding first-order structure (and vice versa). The following is a generalized version of this definition.

Definition 4.2.1. Let ϕ be a formula of L_τ^+, α a formula of L_τ^{FO+}, $a \in \tau(a \neq 0)$ and $\mathfrak{F} = (W, R)$ any τ-frame. We say that ϕ and α are local frame a-correspondents if

$$\mathfrak{F}, w \Vdash_a \phi \text{ iff } \mathfrak{F} \vDash_a \alpha[x = w]$$

where x is any free variable in L_τ^{FO+} and $w \in W$.

In other words, $V(w, \phi) \geq a$ for all valuations V on $\mathfrak{F}$ iff $v(\alpha) \geq a$ for all assignments v such that $v(x) = w$ on the first-order interpretation. The case where $a = 0$ is not included because it is always the case that $V(w, \phi) \geq 0$ and $v(\alpha) \geq 0$. We now define the notion of a *global* frame a-correspondent.

Definition 4.2.2. Let ϕ be a formula of L_τ^+, α a formula of L_τ^{FO+}, $a \in \tau(a \neq 0)$ and $\mathfrak{F} = (W, R)$ any τ-frame. We say that ϕ and α are global frame a-correspondents if

$$\mathfrak{F} \Vdash_a \phi \text{ iff } \mathfrak{F} \vDash_a \alpha$$

where x is any free variable in L_τ^{FO+}.

Example 4.2.3. Let τ be an arbitrary truth-value space. Consider the formula $@_\mathbf{i} p \rightarrow @_\mathbf{i} \Diamond p$ and the first-order formula Rxx (reflexivity). Recall that $@_\mathbf{i} p \rightarrow @_\mathbf{i} \Diamond p$ and Rxx are global frame correspondents in 2-valued setting. We show that $@_\mathbf{i} p \rightarrow @_\mathbf{i} \Diamond p$ and Rxx are also global frame a-correspondents for an arbitrary $a \neq 0 \in \tau$. Let $\mathfrak{F}$ be any τ-frame subject to the reflexive condition and let $w \in W$ be arbitrary. Suppose that Rww, then $\mathfrak{F}, \vDash_a Rxx[x = w]$ for $a \in \tau$ and;

$$\begin{aligned} V(w, @_\mathbf{i} p \rightarrow @_\mathbf{i} \Diamond p) &= V(w, @_\mathbf{i} p) \rightarrow V(w, @_\mathbf{i} \Diamond p) \\ &= V(n(\mathbf{i}), p) \rightarrow V(n(\mathbf{i}), \Diamond p) \\ &= V(n(\mathbf{i}), p) \rightarrow \bigvee \{V(u, p) \wedge Rn(\mathbf{i})u \mid u \in W\} \end{aligned}$$

for an arbitrary valuation V. Since $\mathfrak{F}$ is reflexive, we have that $Rn(\mathbf{i})n(\mathbf{i})$ and therefore,

$V(n(\mathbf{i}), p) \leq \bigvee \{V(u, p) \wedge Rn(\mathbf{i})u \mid u \in W\}$

$\therefore 1 \wedge V(n(\mathbf{i}), p) \leq \bigvee \{V(u, p) \wedge Rn(\mathbf{i})u \mid u \in W\}$

$\therefore 1 \wedge V(w, @_\mathbf{i} p) \leq V(w, @_\mathbf{i} \Diamond p)$

$\therefore 1 \leq V(w, @_{\mathbf{i}}p) \to V(w, @_{\mathbf{i}}\Diamond p)$

Hence, $V(w, @_{\mathbf{i}}p \to @_{\mathbf{i}}\Diamond p) \geq 1 \geq a \in \tau$.

For the implication in the other direction we prove the contrapositive. Suppose that $\neg Rww$ so that $\mathfrak{F} \nVdash_a Rxx[x = w]$ for $a \in \tau$. Let $V(n(\mathbf{i}), p) = 1$ and $V(u, p) = 0$ for all $u \neq n(\mathbf{i})$. Then

$$\begin{aligned} V(w, @_{\mathbf{i}}p \to @_{\mathbf{i}}\Diamond p) &= V(w, @_{\mathbf{i}}p) \to V(w, @_{\mathbf{i}}\Diamond p) \\ &= V(n(\mathbf{i}), p) \to V(n(\mathbf{i}), \Diamond p) \\ &= V(n(\mathbf{i}), p) \to \bigvee\{V(u, p) \wedge Rn(\mathbf{i})u \mid u \in W\} \\ &= 1 \to 0 \\ &= 0 \end{aligned}$$

so that $V(w, @_{\mathbf{i}}p \to @_{\mathbf{i}}\Diamond p) \neq a$ for any $a \neq 0 \in \tau$ at an irreflexive point. Since $v(Rxx) = 1 \geq a$ under any assignment v that assigns a reflexive state w to x, it follows that $@_{\mathbf{i}}p \to @_{\mathbf{i}}\Diamond p$ and Rxx are global frame a-correspondents.

We now give a definition that will be needed in the proof of the correctness of ALBA:

Definition 4.2.4. Given a complex algebra $\mathfrak{F}^+$ and $a \in \tau$, an inequality $\phi \leq \psi$ is *a-true under v in* $\mathfrak{F}^+$ (notation: $\mathfrak{F}^+, v \Vdash_a \phi \leq \psi$) if $v(\phi) \wedge \mathbf{a} \leq v(\psi)$ for the assignment v on $\mathfrak{F}^+$. An inequality $\phi \leq \psi$ is said to be *a-valid in* $\mathfrak{F}^+$ if it is a-true under all assignments v on $\mathfrak{F}^+$ (notation: $\mathfrak{F}^+ \Vdash_a \phi \leq \psi$).

Note that the condition $v(\phi) \wedge \mathbf{a} \leq v(\psi)$ is equivalent to $v(\mathbf{a}) \leq v(\phi) \to v(\psi)$.

The ALBA Algorithm for Many-Valued Hybrid Logic

We now present the version of the many-valued hybrid ALBA algorithm which is a combination of the ALBA algorithms presented in [7] and [13]. We will refer to it as MV-Hybrid-ALBA. With this algorithm we are able to develop a Sahlqvist theory by introducing the class of hybrid inductive formulas. Each hybrid inductive formula is shown to have an effectively computable first-order local frame a-correspondent, where $a \in \tau$ ($a \neq 0$). The algorithm aims to eliminate all propositional variables from a formula or inequality of L^+_τ by applying rules that replace certain syntactic expressions with logically equivalent expressions. When all propositional variables are eliminated, the standard translation is applied to the set of quasi-inequalities produced to obtain a first-order frame a-correspondent for the given formula or inequality. As in [7], we will not invent the new ALBA-like algorithm or even change the old one. This is because ALBA was designed to work on complex algebras that are normal distributive lattice expansions (also known as distributive lattices with operators), see [10] and [11]). Given a τ-frame $\mathfrak{F}$, its complex algebra $\mathfrak{F}^+$ is a perfect Heyting algebra with operators. Hence the ALBA defined in [13] and [7] can be applied to the complex algebra of a τ-frame, subject to the certain conditions which will be given below.

MV-Hybrid ALBA Let $a \in \tau$ and consider the inequality $\phi \leq \psi$. Then the inequality $\phi \wedge \mathbf{a} \leq \psi$ is the ALBA input. The algorithm proceeds as follows:

- Phase 1: Preprocessing

  The aim of this phase is to equivalently break up an inequality $\phi \wedge \mathbf{a} \leq \psi$, given as an input, into smaller inequalities through the application of the rules ($\vee$-Adj) and ($\wedge$-Adj) to be given in the Phase 3. To make it easier, consider the positive generation tree of ϕ and the negative generation tree of ψ, and the surface positive occurrence of $\vee$ and negative occurrence of $\wedge$ by applying the following standard equivalences:

  $$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \qquad \alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
  $$\neg (\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta \qquad \neg (\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$
  $$\Diamond (\alpha \vee \beta) \equiv \Diamond\alpha \vee \Diamond\beta \qquad \Box (\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$$
  $$@_{\mathbf{i}} (\alpha \vee \beta) \equiv @_{\mathbf{i}}\alpha \vee @_{\mathbf{i}}\beta \qquad @_{\mathbf{i}} (\alpha \wedge \beta) \equiv @_{\mathbf{i}}\alpha \wedge @_{\mathbf{i}}\beta$$

  It is easy to check that the inequalities obtained in the preprocessing phase will be of the form $\phi \wedge \mathbf{a} \leq \psi$. Now, let *Preprocess* $(\phi \wedge \mathbf{a} \leq \psi) = \{\phi_i \wedge \mathbf{a} \leq \psi_i \mid i \in I\}$ be the finite set of inequalities obtained after exhaustive application of the above equivalences.

- Phase 2: First approximation

  Each inequality produced in Phase 1 is turned into a quasi-inequality by applying the following *first approximation rule*. The algorithm now proceeds separately on each of the quasi-inequalities obtained.

  First-approximation. Let *Preprocess* $(\phi \wedge \mathbf{a} \leq \psi) = \{\phi_i \wedge \mathbf{a} \leq \psi_i \mid i \in I\}$ be the set of inequalities obtained in Phase 1. Then the following *first-approximation rule* is applied to each $\phi_i \wedge \mathbf{a} \leq \psi_i$ only once:

  $$\frac{\phi_i \wedge \mathbf{a} \leq \psi_i}{\mathbf{j} \leq \phi_i \wedge \mathbf{a} \;\&\; \psi_i \leq \mathbf{m} \Rightarrow \mathbf{j} \leq \mathbf{m}} \text{ (First-approximation)}$$

  where $\mathbf{j} \in J$-VAR and $\mathbf{m} \in M$-VAR.

  The First-approximation yields systems of inequalities $\{\mathbf{j} \leq \phi_i \wedge \mathbf{a} \,\&\, \psi_i \leq \mathbf{m}\}$ for each inequality in *Preprocess* $(\phi \wedge \mathbf{a} \leq \psi)$. Now we apply the ($\wedge$-Adj) rule to obtain $\{\mathbf{j} \leq \phi_i \,\&\, \mathbf{j} \leq \mathbf{a} \,\&\, \psi_i \leq \mathbf{m}\}$. The algorithm will now proceed on $\mathbf{j} \leq \phi$ and $\psi \leq \mathbf{m}$ and leaves $\mathbf{j} \leq \mathbf{a}$ unchanged. Each such system is still called an *initial system*.

- Phase 3: Reduction and Elimination

As in all versions of ALBA algorithm, this phase focuses on eliminating all the propositional variables from the quasi-inequalities (resulting in Phase 2) through the application of the *Ackermann rules* (RH-Ack) and (LH-Ack), or their special case (RH-Ack-0) and (LH-Ack-0). To bring the quasi-inequality into the shape to which one of these rules is applicable, the *approximation*, *residuation* and *adjunction* rules are used. If all propositional variable occurrences have been eliminated, we denote the resulting set of pure quasi-inequalities by $pure\,(\phi \wedge \mathbf{a} \leq \psi)$. If some propositional variable could not be eliminated, then the algorithm fails.

Adjunction rules:

$$\frac{\alpha \leq \beta \wedge \gamma}{\alpha \leq \beta \mathbin{\&} \alpha \leq \gamma}\,(\wedge\text{-Adj}) \qquad \frac{\alpha \vee \beta \leq \gamma}{\alpha \leq \gamma \mathbin{\&} \beta \leq \gamma}\,(\vee\text{-Adj})$$

$$\frac{\alpha \leq \Box\beta}{\blacklozenge\alpha \leq \beta}\,(\Box\text{-Adj}) \qquad \frac{\Diamond\alpha \leq \beta}{\alpha \leq \blacksquare\beta}\,(\Diamond\text{-Adj})$$

$$\frac{\alpha \leq @_{\mathbf{i}}\beta}{@^{\flat}_{\mathbf{i}}\alpha \leq \beta}\,(@\text{-R-Adj}) \qquad \frac{@_{\mathbf{i}}\alpha \leq \beta}{\alpha \leq @^{\#}_{\mathbf{i}}\beta}\,(@\text{-L-Adj})$$

The rules ($\wedge$-Adj) and ($\vee$-Adj) are justified by the fact that $\wedge$ is a right adjoint and $\vee$ is a left adjoint of the diagonal map $\Delta : \mathbf{L} \to \mathbf{A} \times \mathbf{A}$ given by $\Delta(a) = (a, a)$, where $\mathbf{L}$ is a *complex algebra* of the given frame and $\mathbf{A}$ is a subset of $\mathbf{L}$. The rules ($\Box$-Adj) and ($\Diamond$-Adj) are justified by the fact that $\Box$ is the right adjoint of $\blacklozenge$ and $\Diamond$ is the left adjoint of $\blacksquare$, and (@-R-Adj) and (@-R-Adj) follows from the fact that @ operator is the right adjoint of the operator $@^{\flat}$ and @ is also a left adjoint of the operator $@^{\#}$, (see Proposition 2.2.12). Note the last two justifications also hold on a complex algebra of the given frame.

Residuation rules:

$$\frac{\alpha \wedge \beta \leq \gamma}{\alpha \leq \beta \to \gamma}\,(\wedge\text{-Res}) \qquad \frac{\alpha \leq \beta \to \gamma}{\alpha \wedge \beta \leq \gamma}\,(\to\text{-Res})$$

The residuation rules are based on the residuation properties of the interpretations of the connectives.

Approximation rules:

$$\frac{\Box\alpha \leq \mathbf{m}}{\exists \mathbf{m}_0 \left(\Box\mathbf{m}_0 \leq \mathbf{m} \mathbin{\&} \alpha \leq \mathbf{m}_0\right)}\,(\Box\text{-Approx})$$

$$\frac{\mathbf{j} \leq \Diamond\alpha}{\exists \mathbf{j}_0 \left(\mathbf{j} \leq \Diamond \mathbf{j}_0 \mathbin{\&} \mathbf{j}_0 \leq \alpha\right)} \; (\Diamond\text{-Approx})$$

$$\frac{@_{\mathbf{i}}\alpha \leq \mathbf{m}}{\exists \mathbf{m}_0 \left(@_{\mathbf{i}} \mathbf{m}_0 \leq \mathbf{m} \mathbin{\&} \alpha \leq \mathbf{m}_0\right)} \; (@\text{-R-Approx})$$

$$\frac{\mathbf{j} \leq @_{\mathbf{i}}\alpha}{\exists \mathbf{j}_0 \left(\mathbf{j} \leq @_{\mathbf{i}} \mathbf{j}_0 \mathbin{\&} \mathbf{j}_0 \leq \alpha\right)} \; (@\text{-L-Approx})$$

where $\mathbf{m}_0 \in M$-VAR and $\mathbf{j}_0 \in J$-VAR, respectively.

The approximation rules follows from the fact that in a complete and *perfect* heyting algebra each element is the join of join-irreducibles below it and the meet of meet-irreducibles above it.

Ackermann rules: Once the application of the adjunction, residuation and approximation rules has turned the system to the desired shape, the Ackermann rules are applied to the whole system to eliminate all the propositional variables. The Ackermann rules are:

$$\frac{\&_{i=1}^{n} \alpha_i \leq p \mathbin{\&} \&_{j=1}^{m} \beta_j(p) \leq \gamma_j(p)}{\&_{j=1}^{m} \beta_j \left(\bigvee_{i=1}^{n} \alpha_i\right) \leq \gamma_j \left(\bigvee_{i=1}^{n} \alpha_i\right)} \; (\text{RH-Ack})$$

$$\frac{\&_{i=1}^{n} p \leq \alpha_i \mathbin{\&} \&_{j=1}^{m} \gamma_j(p) \leq \beta_j(p)}{\&_{j=1}^{m} \gamma_j \left(\bigwedge_{i=1}^{n} \alpha_i\right) \leq \beta_j \left(\bigwedge_{i=1}^{n} \alpha_i\right)} \; (\text{LH-Ack})$$

where

1. the α_i are p-free;
2. the β_j are positive in p; and
3. the γ_j are negative in p.

If $n = 0$, $\bigvee_{i=0}^{n} \alpha_i \equiv \mathbf{0}$ and $\bigwedge_{i=0}^{n} \alpha_i \equiv \mathbf{1}$, then we have the following special cases (RH-Ack) and (LH-Ack);

$$\frac{\&_{j=1}^{m} \beta_j(p) \leq \gamma_j(p)}{\&_{j=1}^{m} \beta_j(\mathbf{0}) \leq \gamma_j(\mathbf{0})} \; (\text{RH-Ack-0})$$

$$\frac{\&_{j=1}^{m} \gamma_j(p) \leq \beta_j(p)}{\&_{j=1}^{m} \gamma_j(\mathbf{1}) \leq \beta_j(\mathbf{1})} \; (\text{LH-Ack-0})$$

- Phase 4: Translation and Output
  Assuming that it was possible to rewrite an initial system in a form to which one of the Ackermann rules is applicable, else MV-Hybrid-ALBA reports failure and terminates, we denote the set of *pure* quasi-inequalities obtained in Phase 3 by *pure* $(\phi \wedge \mathbf{a} \leq \psi)$, containing no propositional variables. Let $ALBA(\phi \wedge \mathbf{a} \leq \psi)$ be the set of quasi-inequalities:

  $$\underset{i}{\&} \left(pure\left(\phi_i \wedge \mathbf{a} \leq \psi_i\right)\right) \Rightarrow \mathbf{j} \leq \mathbf{m}$$

  for each $\phi_i \wedge \mathbf{a} \leq \psi_i \in Preprocess(\phi \wedge \mathbf{a} \leq \psi)$. All members of $ALBA(\phi \wedge \mathbf{a} \leq \psi)$ are free of propositional variables, so applying the standard translation to each member of $ALBA(\phi \wedge \mathbf{a} \leq \psi)$ will result in a set of first-order correspondents, that is, one for each member of the set of quasi-inequalities. Let $ALBA^{FO}(\phi \wedge \mathbf{a} \leq \psi)$ equal:

  $$\bigwedge_{1 \leq i \leq n} \forall \mathbf{d_i} \forall \mathbf{C_m} \forall \mathbf{c_m} \forall \mathbf{C_j} \forall \vec{\mathbf{d_i}} \forall \vec{\mathbf{c_{m_0}}} \forall \vec{\mathbf{C_{m_0}}} \forall \vec{\mathbf{c_{j_0}}} \forall \vec{\mathbf{C_{j_0}}}$$
  $$(\forall x STx(pure(\phi_i \wedge \mathbf{a} \leq \psi_i)) \Rightarrow \forall x STx(\mathbf{m} \leq \mathbf{j}))$$

  where $\vec{\mathbf{d_i}}, \vec{\mathbf{c_{m_0}}}, \vec{\mathbf{C_{m_0}}}, \vec{\mathbf{c_{j_0}}}$ and $\vec{\mathbf{C_{j_0}}}$ are the vectors of all variables corresponding to the standard translations of nominals, M-VAR and J-VAR, respectively, occurring in $pure(\phi_i \wedge \mathbf{a} \leq \psi_i)$, other than the reserved $\mathbf{m} \in M$-variables and $\mathbf{j} \in J$-variables. Note that the constant variable $\mathbf{c_j}$ corresponding to $\mathbf{j}$ is not quantified over. This is done to produce a local frame correspondent.

Correctness of MV-hybrid-ALBA

The following theorem shows that $\phi \wedge \mathbf{a} \leq \psi$ and $ALBA^{FO}(\phi \wedge \mathbf{a} \leq \psi)$ are local frame correspondents.

Theorem 4.2.5. Let $\mathfrak{F} = (W, R)$ be a τ-frame and let $a \in \tau$. If MV-Hybrid-ALBA succeeds in reducing an inequality $\phi \leq \psi$ and gives out $ALBA^{FO}(\phi \wedge \mathbf{a} \leq \psi)$, then

$$\mathfrak{F}, w \Vdash_t \phi \leq \psi \text{ iff } \mathfrak{F}^+ \left(\mathbf{j} = f_\mathbf{j}\right) \vDash ALBA(\phi \wedge \mathbf{a} \leq \psi) \text{ iff } \mathfrak{F} \vDash ALBA^{FO}\left(\phi \wedge \mathbf{a} \leq \psi\right)\left[\mathbf{c_j} = w\right]$$

where with the $\mathbf{j} \in J$-VAR we associate a function $f_\mathbf{j} \in \tau^W$ such that $f_\mathbf{j}(w) \in J^\infty(\tau)$ for exactly one $w \in W$ and $f_\mathbf{j}(w_0) = 0$ for all $w_0 \neq w$.

Proof. Consider the inequality $\phi \leq \psi$. To complete the proof, the following chain of equivalences must be proven:

$$\mathfrak{F}, w \Vdash_a \phi \leq \psi \text{ iff } \mathfrak{F}^+ \left(\mathbf{j} = f_{\mathbf{j}}\right) \vDash_a \phi \leq \psi \tag{4.1}$$
$$\text{iff } \mathfrak{F}^+ \left(\mathbf{j} = f_{\mathbf{j}}\right) \vDash \phi \wedge \mathbf{a} \leq \psi \tag{4.2}$$
$$\text{iff } \mathfrak{F}^+ \left(\mathbf{j} = f_{\mathbf{j}}\right) \vDash \mathit{Preprocess}(\phi \wedge \mathbf{a} \leq \psi) \tag{4.3}$$
$$\text{iff } \mathfrak{F}^+ \left(\mathbf{j} = f_{\mathbf{j}}\right) \vDash (\mathbf{j} \leq \mathbf{a} \,\&\, \mathbf{j} \leq \phi_i \,\&\, \psi_i \leq \mathbf{m}) \Rightarrow (\mathbf{j} \leq \mathbf{m}) \tag{4.4}$$
$$\text{iff } \mathfrak{F}^+ \left(\mathbf{j} = f_{\mathbf{j}}\right) \vDash \mathit{pure}(\phi_i \wedge \mathbf{a} \leq \psi_i) \Rightarrow (\mathbf{j} \leq \mathbf{m}) \tag{4.5}$$
$$\text{iff } \mathfrak{F} \vDash \mathit{ALBA}\,(\phi \wedge \mathbf{a} \leq \psi) \tag{4.6}$$
$$\text{iff } \mathfrak{F} \vDash \mathit{ALBA}^{FO}\,(\phi \wedge \mathbf{a} \leq \psi)\left[\mathbf{c}_{\mathbf{j}} = w\right] \tag{4.7}$$

(4.1): Follows from Proposition 2.3.6.
(4.1) iff (4.2): Follows from Definition 4.2.4.
(4.2) iff (4.3): It was shown in [7] that application of the preprocessing rules will result in inequalities of the form $\phi_i \wedge \mathbf{a} \leq \psi_i$ and the preprocessing rules preserve truth.
(4.3) iff (4.4): The proof of this equivalence is found in [10] and [7].
(4.4) iff (4.5): The reduction rules need not be applied to the inequality $\mathbf{j} \leq \mathbf{a}$, as it is already propositional variable free. By proofs of the correctness of ALBA in [10] and [7], the reduction rules (containing the modal operators) preserve truth of the inequalities $\mathbf{j} \leq \phi_i$ and $\psi \leq \mathbf{m}$. We will extend the proofs in [10] and [7] by showing the reduction rules containing hybrid operators also preserve the truth of the inequalities $\mathbf{j} \leq \phi_i$ and $\psi \leq \mathbf{m}$. We want to show that the following rules

1. $$\frac{@_{\mathbf{i}}\phi \leq \psi}{\phi \leq @^{\#}_{\mathbf{i}}\psi}$$

2. $$\frac{\phi \leq @_{\mathbf{i}}\psi}{@^{\flat}_{\mathbf{i}}\phi \leq \psi}$$

3. $$\frac{\mathbf{j} \leq @_{\mathbf{i}}\phi}{\mathbf{j} \leq @_{\mathbf{i}}\mathbf{j_0} \,\&\, \mathbf{j_0} \leq \phi}$$

4. $$\frac{@_{\mathbf{i}}\phi \leq \mathbf{m}}{@_{\mathbf{i}}\mathbf{m_0} \leq m \,\&\, \phi \leq \mathbf{m_0}}$$

are sound. That is, the premise is true in a complex algebra under an assignment if and only if the conclusion is true in the same algebra under the same assignment.

1. Follows from the fact that @ is a left adjoint of $@^{\#}$.

2. Follows from the fact that @ is a right adjoint of $@^{\flat}$.

3. Let $\mathfrak{F}^+$ be a complex algebra of the given τ-frame $\mathfrak{F} = (W, R)$ and let v be an assignment on $\mathfrak{F}^+$. Suppose that $v(\mathbf{j}) \leq v(@_\mathbf{i}\phi)$ Then we have that $v(\mathbf{j}) \leq @_{v(\mathbf{i})}(v(\phi))$. Since $J^\infty(\tau^W)$ join generates τ^W, we have that $v(\mathbf{j}) \leq @_{v(\mathbf{i})} \bigvee\{\mathbf{j}' \in J^\infty(\tau^W) \mid \mathbf{j}' \leq v(\phi)\}$. Hence $v(\mathbf{j}) \leq \bigvee\{@_{v(\mathbf{i})}\mathbf{j}' \mid J^\infty(\tau^W) \ni \mathbf{j}' \leq v(\phi)\}$ since $@_{v(\mathbf{i})}$ is completely join-preserving (see Lemma 2.3.2). Therefore $v(\mathbf{j}) \leq @_{v(\mathbf{i})}\mathbf{j_0}$ for some $\mathbf{j_0} \in J^\infty(\tau^W)$ such that $\mathbf{j_0} \leq v(\phi)$. Let $v' \sim_{\mathbf{j_0}} v$ be such that $v'(\mathbf{j}) = \mathbf{j_0}$. Note that we are using the fact that $\mathbf{j_0}$ is a new variable in J-VAR. Then $v'(\mathbf{j}) \leq v'(@_\mathbf{i}\mathbf{j_0})$ and $v'(\mathbf{j_0}) \leq v'(\phi)$. Conversely, suppose that $v(\mathbf{j}) \leq v(@_\mathbf{i}\mathbf{j_0})$ and $v(\mathbf{j_0}) \leq v(\phi)$. Then $v(\mathbf{j}) \leq @_{v(\mathbf{i})}v(\mathbf{j_0})$ and $v(\mathbf{j_0}) \leq v(\phi)$. So, by the monotonicity of $@_{v(\mathbf{i})}$, we have that,
$$v(\mathbf{m}) \leq @_{v(\mathbf{i})}\left(v(\mathbf{j_0})\right) \leq @_{v(\mathbf{i})}\left(v(\phi)\right) = v(@_\mathbf{i}\phi).$$

4. Let $\mathfrak{F}^+$ be a complex algebra of the given τ-frame $\mathfrak{F} = (W, R)$ and let v be an assignment on $\mathfrak{F}^+$. Suppose that $v(@_\mathbf{i}\phi) \leq v(\mathbf{m})$. Then we have that $@_{v(\mathbf{i})}\left(v(\phi)\right) \leq v(\mathbf{m})$. Since $M^\infty(\tau^W)$ meet generates τ^W, we have that $@_{v(\mathbf{i})} \bigwedge\{\mathbf{m}' \in M^\infty(\tau^W) \mid v(\phi) \leq \mathbf{m}'\} \leq v(\mathbf{m})$. Therefore, $\bigwedge\{@_{v(\mathbf{i})}\mathbf{m}' \mid v(\phi) \leq \mathbf{m}' \in M^\infty(\tau^W)\} \leq v(\mathbf{m})$ since $@_{v(\mathbf{i})}$ is completely meet-preserving (see Lemma 2.3.2). Hence, it follows that $@_{v(\mathbf{i})}\mathbf{m_0} \leq v(\mathbf{m})$ for some $\mathbf{m_0} \in M^\infty(\tau^W)$ such that $v(\phi) \leq \mathbf{m_0}$. Let $v' \sim_{\mathbf{m_0}} v$ be such that $v'(\mathbf{m}) = \mathbf{m_0}$. Note we are using the fact that $\mathbf{m_0}$ is a new variable in M-VAR. Then
$$v'(@_\mathbf{i}\mathbf{m_0}) \leq v'(\mathbf{m}) \text{ and } v'(\phi) \leq v'(\mathbf{m_0})$$
Conversely, suppose that $v(@_\mathbf{i}\mathbf{m_0}) \leq v(\mathbf{m})$ and $v(\phi) \leq v(\mathbf{m_0})$. Hence,
$$@_{v(\mathbf{i})}\left(v(\mathbf{m_0})\right) \leq v(\mathbf{m}) \text{ and } v(\phi) \leq v(\mathbf{m_0}).$$
So, by monotonicity of $@_{v(\mathbf{i})}$,
$$@_{v(\mathbf{i})}\left(v(\phi)\right) \leq @_{v(\mathbf{i})}\left(v(\mathbf{m_0})\right) \leq v(\mathbf{m}).$$
Therefore, $v(@_\mathbf{i}\phi) \leq v(\mathbf{m})$.

(4.5) iff (4.6): Follows from the definition of ALBA($\phi \wedge \mathbf{a} \leq \psi$), which is defined as:
$$\&(pure(\phi_i \wedge \mathbf{a} \leq \psi_i)) \Rightarrow \mathbf{j} \leq \mathbf{m}$$
for each $\phi_i \wedge \mathbf{a} \leq \psi_i \in Preprocess(\phi \wedge \mathbf{a} \leq \psi)$.
(4.6) iff (4.7): Follows from a straightforward extension of Proposition 4.1.20 to cover quasi-inequalities. □

Theorem 4.2.6. MV-Hybrid ALBA succeeds on the class of inductive inequalities (and hence the class of Sahlqvist inequalities) of L^+_τ over all τ-frames.

Corollary 4.2.7. Let τ be any perfect Heyting algebra and let $a \in \tau$. All inductive formulas have effectively computable first-order local frame a-correspondents over the class of all τ-frames.

We now have a sufficient condition that assures us that an inequality will have a first-order frame correspondent and that condition is that the inequality be inductive. Note that this is a very strong Sahlqvist type result, as it is applicable to a wide range of truth-value spaces.

We now consider examples to see how the MV-Hybrid-ALBA algorithm is utilized to find the first-order correspondent of the given hybrid inequality.

Example 4.2.8. The following example is also found in [7]. We will do it differently here to illustrate the importance of further simplifying the *pure formula* (formula with eliminated propositional variables) before applying the standard translation. We know that in the 2-valued and many-valued setting the formula $\phi = p \rightarrow \Diamond p$ and Rxx are each other's local frame a-correspondents where $a \in \tau$ $(a \neq 0)$. Now, let $a \in \tau$ $(a \neq 0)$. The corresponding in inequality $a \wedge p \leq \Diamond p$ remains unchanged under preprocessing and first-approximation turns in into

$$\& \{\mathbf{j} \leq a \wedge p\, ,\, \Diamond p \leq \mathbf{m}\} \Rightarrow \mathbf{j} \leq \mathbf{m}$$

Now applying $(\wedge$-Adj$)$ to $\mathbf{j} \leq a \wedge p$ gives

$$\& \{\mathbf{j} \leq a\, ,\, \mathbf{j} \leq p\, ,\, \Diamond p \leq \mathbf{m}\} \Rightarrow \mathbf{j} \leq \mathbf{m}$$

Applying (RH-Ack) gives

$$\& \{\mathbf{j} \leq a\, ,\, \Diamond \mathbf{j} \leq \mathbf{m}\} \Rightarrow \mathbf{j} \leq \mathbf{m}$$

At this point all propositional variables have been eliminated and the standard translation of the above quasi-inequality will be a first-order frame correspondent of this formula. Before attempting this translation, however, we first simplify the quasi-inequality. Applying $(\Diamond$-Adj$)$ on $\Diamond \mathbf{j} \leq \mathbf{m}$ we have

$$\& \{\mathbf{j} \leq a\, ,\, \mathbf{j} \leq \blacksquare \mathbf{m}\} \Rightarrow \mathbf{j} \leq \mathbf{m}.$$

Now, since $\mathbf{j}$ is less than or equals to both a and $\blacksquare \mathbf{m}$, it follows that

$$\mathbf{j} \leq a \wedge \blacksquare \mathbf{m} \Rightarrow \mathbf{j} \leq \mathbf{m}.$$

The above implication implies that

$$a \wedge \blacksquare \mathbf{m} \leq \mathbf{m}.$$

Now, we apply the standard translation to obtain the following:

$STx(a \wedge \blacksquare \mathbf{m}) \leq STx(\mathbf{m})$

$\equiv STx(a) \wedge STx(\blacksquare \mathbf{m}) \leq STx(\mathbf{m})$

$\equiv a \wedge \forall y(Ryx \rightarrow STy(\mathbf{m})) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$

$\equiv a \wedge \forall y(Ryx \rightarrow (y \neq \mathbf{c_m}) \vee \mathbf{C_m}) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$

Now, we write out the translation in full detail:

$$\forall x \forall \mathbf{c_m} \forall \mathbf{C_m} \, [a \wedge \forall y(Ryx \rightarrow (y \neq \mathbf{c_m}) \vee \mathbf{C_m}) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}]$$

We have two cases to consider. The case where $x \neq \mathbf{c_m}$ is not interesting because the right-hand side of the inequality is 1 and hence makes the whole inequality 1. The case $x = \mathbf{c_m}$ is the one that will give us the corresponding first-oder formula. Now, we have that

$\forall x \forall \mathbf{C_m} \, [a \wedge \forall y(Ryx \rightarrow (y \neq x) \vee \mathbf{C_m}) \leq \mathbf{C_m}]$

$\equiv \forall x \forall \mathbf{C_m} \, [a \wedge (Rxx \rightarrow \mathbf{C_m}) \leq \mathbf{C_m}]$

$\equiv \forall x \forall \mathbf{C_m} \, [a \leq (Rxx \rightarrow \mathbf{C_m}) \rightarrow \mathbf{C_m}]$

$\equiv \forall x(a \leq \forall \mathbf{C_m} \, [(Rxx \rightarrow \mathbf{C_m}) \rightarrow \mathbf{C_m}])$

$\equiv \forall x(a \leq Rxx).$

Example 4.2.9. We now use the algorithm to get the first-order local frame a-correspondent formula of the hybrid formula $\phi = @_{\mathbf{i}} p \rightarrow @_{\mathbf{j}} \Diamond p$. The corresponding inequality $\phi = @_{\mathbf{i}} p \leq @_{\mathbf{j}} \Diamond p$ remains unchanged under preprocessing and first-approximation turns it into

$$\& \{\mathbf{j_0} \leq a \wedge @_{\mathbf{i}} p \, , \, @_{\mathbf{j}} \Diamond p \leq \mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Applying ($\wedge$-Adj) to $\mathbf{j_0} \leq a \wedge @_{\mathbf{i}} p$ gives

$$\& \{\mathbf{j_0} \leq a \, , \, \mathbf{j_0} \leq @_{\mathbf{i}} p \, , \, @_{\mathbf{j}} \Diamond p \leq \mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Applying (@-R-Adj) on $\mathbf{j_0} \leq @_{\mathbf{i}} p$ gives

$$\& \{\mathbf{j_0} \leq a \, , \, @^{\flat}_{\mathbf{i}} \mathbf{j_0} \leq p \, , \, @_{\mathbf{j}} \Diamond p \leq \mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Now applying (RH-Ack) we obtain

$$\& \{\mathbf{j_0} \leq a \, , \, @_{\mathbf{j}} \Diamond @^{\flat}_{\mathbf{i}} \mathbf{j_0} \leq \mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

At this point all propositional variables have been eliminated and the standard translation of the above quasi-inequality will be a first-order frame correspondent of this formula. Before attempting this translation, however, we first simplify the quasi-inequality. Applying (@-L-Adj) to $@_{\mathbf{j}}\Diamond @^{\flat}_{\mathbf{i}}\mathbf{j_0}$ we get

$$\&\{\mathbf{j_0} \leq a\,,\ \Diamond @^{\flat}_{\mathbf{i}}\mathbf{j_0} \leq @^{\#}_{\mathbf{j}}\mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Applying ($\Diamond$-Adj) on $\Diamond @^{\flat}_{\mathbf{i}}\mathbf{j_0} \leq @^{\#}_{\mathbf{j}}\mathbf{m_0}$ we obtain

$$\&\{\mathbf{j_0} \leq a\,,\ @^{\flat}_{\mathbf{i}}\mathbf{j_0} \leq \blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Applying (@-R-Adj) on $@^{\flat}_{\mathbf{i}}\mathbf{j_0} \leq \blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}$ gives

$$\&\{\mathbf{j_0} \leq a\,,\ \mathbf{j_0} \leq @_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}\} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

Now since $\mathbf{j_0}$ is less than or equal to both a and $@_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}$, it follows that

$$\mathbf{j_0} \leq a \wedge @_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0} \Rightarrow \mathbf{j_0} \leq \mathbf{m_0}$$

The above implication implies that

$$a \wedge @_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0} \leq \mathbf{m_0}$$

Now we take the standard translation of the simplified pure formula. That is,

$STx(a \wedge @_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}) \leq STx(\mathbf{m_0})$

$\equiv STx(a) \wedge STx(@_{\mathbf{i}}\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0}) \leq STx(\mathbf{m_0})$

$\equiv a \wedge \exists y(y = \mathbf{d_i} \wedge STy(\blacksquare @^{\#}_{\mathbf{j}}\mathbf{m_0})) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$

$\equiv a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow STz(@^{\#}_{\mathbf{j}}\mathbf{m_0}))) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$

$\equiv a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow (z = \mathbf{d_j}) \rightarrow \forall t(STt))) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$

$\equiv a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow (z = \mathbf{d_j}) \rightarrow \forall t((t \neq \mathbf{c_m}) \vee \mathbf{C_m}))) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}$
Now, we write the translation in full detail:

$$\forall x \forall \mathbf{c_m} \forall \mathbf{C_m} \forall \mathbf{d_i} \forall \mathbf{d_j}$$
$$[a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow (z = \mathbf{d_i}) \rightarrow \forall t((t \neq \mathbf{c_m}) \vee \mathbf{C_m}))) \leq (x \neq \mathbf{c_m}) \vee \mathbf{C_m}]$$

Now we have two cases to consider. The case where $x \neq \mathbf{c_m}$ is equal to 1 because the right-hand side of the inequality becomes 1 and hence making the whole formula 1. So, the case where $x = \mathbf{c_m}$ is the one that will give us the corresponding first-order formula. So, we have that:

$$\forall \mathbf{C_m} \forall \mathbf{d_i} \forall \mathbf{d_j} \, [a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow (z = \mathbf{d_i}) \rightarrow \forall t((t \neq \mathbf{c_m}) \vee \mathbf{C_m}))) \leq \mathbf{C_m}]$$

$$\equiv \forall \mathbf{C_m} \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \wedge \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow ((z = \mathbf{d_j}) \rightarrow \mathbf{C_m}))) \leq \mathbf{C_m}\right]$$

$$\equiv \forall \mathbf{C_m} \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq \exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow ((z = \mathbf{d_j}) \rightarrow \mathbf{C_m}))) \rightarrow \mathbf{C_m}\right]$$

$$\equiv \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq \forall \mathbf{C_m}(\exists y(y = \mathbf{d_i} \wedge \forall z(Rzy \rightarrow ((z = \mathbf{d_j}) \rightarrow \mathbf{C_m})))) \rightarrow \mathbf{C_m}\right]$$

$$\equiv \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq \forall \mathbf{C_m}(\forall z(Rz\mathbf{d_i} \rightarrow ((z = \mathbf{d_j}) \rightarrow \mathbf{C_m}))) \rightarrow \mathbf{C_m}\right]$$

$$\equiv \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq \forall \mathbf{C_m}(\forall z(Rz\mathbf{d_i} \wedge z = \mathbf{d_j}) \rightarrow \mathbf{C_m}) \rightarrow \mathbf{Cm}\right]$$

$$\equiv \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq \forall \mathbf{C_m}(R\mathbf{d_i}\mathbf{d_j} \rightarrow \mathbf{C_m}) \rightarrow \mathbf{C_m}\right]$$

$$\equiv \forall \mathbf{d_i} \forall \mathbf{d_j} \left[a \leq R\mathbf{d_i}\mathbf{d_j}\right]$$

Conclusion

In this work we managed to prove t-invariance results for generated submodels (Theorem 3.1.2) and t-invariance results for bounded morphisms (Theorem 3.2.3) for many-valued hybrid logic. We have also proved that t-bisimilarity implies hybrid t-equivalence in general (Theorem 3.4.3). However, the converse is not true in general. We proved that the converse is true for a *weaker* notion of a bisimulation for a special set of models, *image t-finite models* (Theorem 3.4.11). Lastly, we have proved Theorem 4.2.5 which is the correctness of MV-Hybrid ALBA. A consequence of Theorem 4.2.5 asserts that the class of *inductive formulas* always have a local frame correspondent subject only to the condition that the truth-value space is a perfect Heyting algebra (Corollary 4.2.7).

There is still much to be done in the expressivity of many-valued hybrid logic. One of the obvious goal would be to obtain a Goldblatt–Thomason style theorem for many-valued modal logic and many-valued hybrid logic. The expressivity considered in this dissertation is building up to the Goldblatt–Thomason style theorem for many-valued hybrid logic.

Sahlqvist theory has two main parts: the correspondence and canonicity. The correspondence theory for many-valued modal logic was considered in [7]. The work in this dissertation also considered the correspondence part of the Sahlqvist theory for many-valued hybrid logic. The canonicity part of Sahlqvist's theorem is still outstanding for both these logics.

The properties of many-valued hybrid logic remain mostly unexplored. For example, it is currently unknown whether the logic enjoys *interpolation* or *Beth definability*. Another possible study would be to obtain a complete *Hilbert-style axiomatization* of the minimal many-valued hybrid logic.

References

[1] C. Areces, P. Blackburn, and M. Marx, *The computational complexity of hybrid temporal logics*, Logic Journal of the IGPL, 8 (2000), pp. 653–679.

[2] C. E. Areces, *Logic engineering: the case of description and hybrid logics*, PhD thesis, University of Amsterdam, 2000.

[3] P. Blackburn, M. De Rijke, and Y. Venema, *Modal Logic*, vol. 53, Cambridge University Press, 2002.

[4] P. Blackburn and J. Seligman, *Hybrid languages*, Journal of Logic, Language and Information, 4 (1995), pp. 251–272.

[5] P. Blackburn and M. Tzakova, *Hybridizing concept languages*, Annals of Mathematics and Artificial Intelligence, 24 (1998), pp. 23–49.

[6] ——, *Hybrid languages and temporal logic*, Logic Journal of the IGPL, 7 (1999), pp. 27–54.

[7] C. Britz, *Correspondence theory in many-valued modal logic*, Master's thesis, University of Johannesburg, South Africa, 2016. `https://ujcontent.uj.ac.za/vital/access/services/Download/uj:24102/SOURCE1?view=true)`.

[8] R. Bull, *An approach to tense logic*, Theoria, 36 (1970), pp. 282–300.

[9] S. Burris and H. P. Sankappanavar, *A Course in Universal Algebra*, 2006.

[10] W. Conradie and A. Palmigiano, *Algorithmic correspondence and canonicity for distributive modal logic*, Annals of Pure and Applied Logic, 163 (2012), pp. 338–376.

[11] ——, *Algorithmic correspondence and canonicity for non-distributive logics*, Annals of Pure and Applied Logic, 170 (2019), pp. 923–974.

[12] W. Conradie, A. Palmigiano, S. Sourabh, and Z. Zhao, *Canonicity and relativized canonicity via pseudo-correspondence: an application of ALBA*, arXiv preprint arXiv:1511.04271, (2015).

[13] W. Conradie and C. Robinson, *On Sahlqvist theory for hybrid logics*, Journal of Logic and Computation, 27 (2017), pp. 867–900.

[14] B. A. DAVEY AND H. A. PRIESTLEY, *Introduction to lattices and order*, Cambridge University Press, 2002.

[15] J. M. DUNN, M. GEHRKE, AND A. PALMIGIANO, *Canonical extensions and relational completeness of some substructural logics*, The Journal of Symbolic Logic, 70 (2005), pp. 713–740.

[16] P. E. ELEFTHERIOU AND C. D. KOUTRAS, *Frame constructions, truth invariance and validity preservation in many-valued modal logic*, Journal of Applied Non-Classical Logics, 15 (2005), pp. 367–388.

[17] P. E. ELEFTHERIOU, C. D. KOUTRAS, AND C. NOMIKOS, *Notions of bisimulation for Heyting-valued modal languages*, Journal of Logic and Computation, 22 (2012), pp. 213–235.

[18] M. FITTING, *Many-valued modal logics*, Fundam. Inform., 15 (1991), pp. 235–254.

[19] N. GALATOS, P. JIPSEN, T. KOWALSKI, AND H. ONO, *Residuated lattices: an algebraic glimpse at substructural logics*, vol. 151, Elsevier, 2007.

[20] V. GORANKO AND D. VAKARELOV, *Elementary canonical formulae: extending Sahlqvist's theorem*, Annals of Pure and Applied Logic, 141 (2006), pp. 180–217.

[21] J. HANSEN, T. BOLANDER, AND T. BRAÜNER, *Many-valued hybrid logic*, in Advances in modal logic (Vol. 7), C. Areces and R. Goldblatt, eds., College Publications, 2008, pp. 111–132.

[22] R. E. HODEL, *An introduction to mathematical logic*, Dover Publications, 2013.

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

[24] O. MORIKAWA, *Some modal logics based on a three-valued logic.*, Notre Dame Journal of Formal Logic, 30 (1988), pp. 130–137.

[25] P. OSTERMANN, *Many-valued modal propositional calculi*, Mathematical Logic Quarterly, 34 (1988), pp. 343–354.

[26] A. N. PRIOR, *Past, present and future*, vol. 154, Clarendon Press Oxford, 1967.

[27] H. SAHLQVIST, *Correspondence and completeness in the first and second order semantics for modal logic*, in Proceedings of the 3rd Scandinavian Logic Symposium (Uppsala 1973), S. Kanger, ed., North-Holland, 1975, pp. 110–143.

[28] P. K. SCHOTCH, J. B. JENSEN, P. F. LARSEN, AND E. J. MACLELLAN, *A note on three-valued modal logic.*, Notre Dame Journal of Formal Logic, 19 (1978), pp. 63–68.

[29] B. D. ten Cate, *Model theory for extended modal languages*, PhD thesis, University of Amsterdam, 2005. http://www.illc.uva.nl/Research/Publications/Dissertations/DS-2005-01.text.pdf.

[30] S. Thomason, *Possible worlds and many truth values*, Studia Logica, 37 (1978), pp. 195–204.

[31] J. van Benthem, *Modal frame correspondences and fixed-points*, Studia Logica, 83 (2006), pp. 133–155.