Extensions of Modal Logic KTB and Other Topics

Michael Stevens

August 2009

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

Dedicated to my parents.

Declaration

The work in this thesis is my own except where otherwise stated.

Michael Stevens

Acknowledgements

I would like to thank Tomasz Kowalski, my first supervisor, for his support during the first couple of years of my work. He was always available for discussion, and his help was invaluable.

I also thank Rajeev Gore, who stepped into the supervisor role after Tomasz left Australia, and did an impressive job, in spite of his unfamiliarity with my prior work. I'd also like to thank Tomasz's colleagues, particularly Yutaka Miyazaki, who provided me with some of Tomasz's notes, enabling me to reconstruct some of our work after his departure.

Thanks to John Lloyd and John Slaney. Both have taken time to provide useful advice and input at various points. John Lloyd in particular was so good as to read an earlier draft of this thesis, and provide comments.

Abstract

This thesis covers four topics. They are the extensions of the modal logic **KTB**, the use of normal forms in modal logic, automated reasoning in the modal logic **S4** and the problem of unavoidable words.

Extensions of KTB: The modal logic **KTB** is the logic of reflexive and symmetric frames. Dually, KTB-algebras have a unary (normal) operator f that satisfies the identities $f(x) \geq x$ and $\neg x \leq f(\neg f(x))$. Extensions of **KTB** are subvarieties of the algebra KTB. Both of these form a lattice, and we investigate the structure of the bottom of the lattice of subvarieties. The unique atom is known to correspond to the modal logic whose frame is a single reflexive point. Yutaka demonstrated that this atom has a unique cover, corresponding to the frame of the two element chain. We construct covers of this element, and so demonstrate that there are a continuum of such covers.

Normal Forms in Modal Logic: Fine proposed the use of normal forms as an alternative to traditional methods of determining Kripke completeness. We expand on this paper and demonstrate the application of normal forms to a number of traditional modal logics, and define new terms needed to apply normal forms in this situation.

Automated reasoning in S4: History based methods for automated reasoning are well understood and accepted. Pliuškevičius & Pliuškevičienė propose a new, potentially revolutionary method of applying marks and indices to sequents. We show that the method is flawed, and empirically compare a different mark/index based method to the traditional methods instead.

Unavoidable words: The unavoidable words problem is concerned with repetition in strings of symbols. There are two main ways to identify a word as unavoidable, one based on generalised pattern matching and one from an algorithm. Both methods are in NP, but do not appear to be in P. We define the simple unavoidable words as a subset of the standard unavoidable words that can be identified by the algorithm in P-time. We define depth separating

homomorphisms as an easy way to generate a subset of the unavoidable words using the pattern matching method. We then show that the two simpler problems are equivalent to each other.

Contents

Chapter 1

Introduction

Modal logic is an extension of classical logic. Where classical logic merely states things as true or false, using syntactic connectives for conjunction ($\wedge$), disjunction ($\vee$), implication ($\rightarrow$) and negation ($\neg$), modal logic extends this to handle concepts that are *possibly* true or false, with extra connectives for necessity ($\Box$) and possibility ($\Diamond$). This thesis covers a number of disparate topics, but modal logic forms a unifying thread for most of the various chapters.

While classical logic uses truth tables for its semantics, the semantics of modal logic is based on directed graphs, called frames. Intuitively, every vertex in a graph can give a different truth value to logical propositions. The normal modal logic **K** is based on the set of all possible graphs, but there are many modal logics, each based on a different set of graphs. For example, **KTB** is based on reflexive, symmetric graphs, while **S4** is based on reflexive, transitive graphs.

Axiomatically, classical logic uses a finite set of axioms and the rule of Modus Ponens. To this, the modal logic **K** adds a new rule of inference, the rule of necessitation, which allows our reasoning to extend from a single vertex to many. As with classical logics, one may characterise individual modal logics by extending a base modal logic with axioms. The logic **KTB** is the logic **K** extended by axioms $\Box p \rightarrow p$ (reflexivity), and $\Diamond\Box p \rightarrow p$ (symmetry), while **S4** is **K** extended by the axioms $\Box p \rightarrow p$ (reflexivity) and $\Box p \rightarrow \Box\Box p$ (transitivity).

Classical logic has well defined normal forms, such as disjunctive normal form and conjunctive normal form. It is possible to transform any formula into one of these normal forms without changing its logical validity. If a formula is true under a particular set of circumstances, then the normal form of the formula will also be true, and vice versa. Modal logic, being based on classical logic, inherits this property, albeit with more complex normal forms.

While humans can work with modal logics with relative ease, computers require strict rules. Automated reasoning focuses on the proof methods that allow computers to create a proof within the logic without human input. Usually, we care about satisfiability proofs, showing that under some set of circumstances, a formula is true, and that these circumstances are compatible with the assumptions of the logic under discussion.

Each chapter of this thesis covers a different field. One chapter discusses extensions of the modal logic **KTB**, one discusses normal forms in modal logics, one discusses the automated reasoning methods for the modal logic **S4**, and a last covers unavoidable words. Each chapter has been made as self-contained as possible. It is possible to read and understand an individual chapter without touching the other three.

KTB is in some sense, a logic of graphs, so the question of its possible extensions is an interesting one. It is well known that the logic, like all reflexive logics, has a single maximal extension. The problem of almost maximal extensions, in the sense that any further specification will reach the maximal extension, has also been solved. Chapter 3 discusses the logical next step from these two results, the case of logics that, if extended, will reach an "almost-maximal" extension, and shows that while the previous two classes were small, this class is uncountably large.

While our intuition of modal logics is that they reason within directed frames, some logics are not so simple. The property of being defined by a set of frames, as **KTB** is defined by reflexive, transitive frames, is known as Kripke completeness. When discussing modal logics, the problem of Kripke completeness is essentially a practical one. Kripke frames are a powerful and intuitive tool for reasoning about logics. However, Kripke frames can only be applied to Kripke complete logics. Thus, proving Kripke completeness is an important step in applications of modal logic. There are a couple of methods that are popularly used to prove that a logic has Kripke completeness. Chapter 4 discusses a third, less well known method of demonstrating completeness.

Automated reasoning is a reasonably mature field, with applications in artificial intelligence[35], computer security[11] and other areas[4]. For automated reasoning, it is necessary to guarantee termination since it is possible for a computer to enter an infinite loop, applying the same rules over and over. The standard, history-based methods of ensuring termination keep track of rule application and prevent repetition. They are well understood. However, the paper [36] claims a new method, that could potentially be dramatically more efficient

than the existing methods. Chapter 5 discusses this new method, and how it compares to the existing methods.

Chapter 6 is not about modal logic. Instead, we consider the problem of unavoidable words, which has applications to assorted algebraic problems, as was first discovered in 1906. Loosely, it involves determining that some patterns *must* occur in any sufficiently long string, while other patterns can be avoided by strings of arbitrary length.

The obvious question that arises from such a problem is determining which patterns are unavoidable, and which may be avoided. In the 1970's an algorithm was discovered to solve this problem. However, the algorithm is complex. Chapter 6 discusses some properties of the problem, and how they might be used to simplify the algorithm.

Chapter 2

Logical and Algebraic preliminaries

Some of the chapters in this thesis use common definitions of logic and algebra. For convenience, these definitions are placed here, in a separate chapter, rather than being repeated in each chapter. Readers familiar with modal logic and algebra may skip this section. Readers seeking a more detailed introduction to modal logic are advised to consult a textbook. The logical introduction is based mostly on the book Modal Logic, by Chagrov and Zakharyaschev [10]. An exellent introduction to algebraic logic is provided by Venema [7]

2.1 Formulas

To begin with, we shall define formulas. A formula, usually written with lower case Greek letters φ, ψ, χ, is made up of some basic elements, bound together with some set of logical connectives. A set of formulas will be denoted with uppercase Greek letters, such as Γ, Σ, Δ. We shall commence with a purely syntactic definition:

The first basic element of a formula is its set of **propositional variables**, denoted with lower case Roman letters $p, q, r \ldots$, sometimes with subscripts or, more rarely, superscripts for distinguishing multiple variables. The second basic element is the logical constant of falsehood, $\bot$.

To create a formula, these basic elements are bound together with binary logical connectives $\wedge$ (and), $\vee$ (or), $\rightarrow$ (implies), and the punctuation of "(" and ")". Thus, we recursively define the set of formulas:

As a base case, $\bot$ is a formula, as is any propositional variable p.

For the recursion, if φ and ψ are both formulas, then so are:

$$\varphi \wedge \psi \quad \varphi \dot{\wedge} \psi \quad \varphi \to \psi \quad (\varphi)$$

To this, we add a few derived elements. The unary logical operator $\neg$ (not) is defined by $\neg\varphi = (\varphi \to \bot)$. The logical constant of truth, $\top$, is defined by being not falsehood: $\top = \neg\bot = (\bot \to \bot)$.

2.1.1 Semantics

Having defined the syntax of formulas, it is reasonable to ask what they actually mean. A **valuation** is a function that maps propositional variables to true/false values. A valuation v can be used to construct a recursive function f_v that maps formulas to truth values, as follows:

$$f_v(\bot) = \text{false}$$

$$f_v(p) = v(p)$$

$$f_v(\varphi \wedge \psi) = \begin{cases} \text{true} & \text{if } f_v(\varphi) = \text{true and } f_v(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_v(\varphi \vee \psi) = \begin{cases} \text{true} & \text{if } f_v(\varphi) = \text{true or } f_v(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_v(\varphi \to \psi) = \begin{cases} \text{true} & \text{if } f_v(\varphi) = \text{false or } f_v(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

From this definition, we can derive f_v for our derived operators:

$$f_v(\top) = \text{true}$$

$$f_v(\neg\varphi) = \begin{cases} \text{true} & \text{if } f_v(\varphi) = \text{false} \\ \text{false} & \text{if } f_v(\varphi) = \text{true} \end{cases}$$

If φ is true for some particular v_1, we say that v_1 provides a model for φ. If φ is false for a valuation v_2, we say v_2 provides a counter-model for φ, or that v_2 refutes φ. We say that a formula φ is **valid** if for all possible v we have $f_v(\varphi) = \text{true}$.

2.1.2 Substitution

The operation of substitution transforms formulas into other formulas. A **substitution s** is a set. Its elements have the form φ/p, where φ is a formula and p a propositional variable. In a substitution, each propositional variable occurs at most once on the right hand side: it cannot have both φ/p and ψ/p as members. From the substitution $\mathbf{s}$, we define a function s from propositional variables to formulae:

$$s(p) = \begin{cases} \varphi & \text{if } \varphi/p \in \mathbf{s} \\ p & \text{otherwise} \end{cases}$$

Given this function, we define the operation $\varphi\mathbf{s}$ of applying a substitution to a formula as follows:

As a base:

$$p\mathbf{s} = s(p)$$

$$\bot\mathbf{s} = \bot$$

For the recursive case, for any binary operation $\odot$,

$$(\varphi \odot \psi)\mathbf{s} = \varphi\mathbf{s} \odot \psi\mathbf{s}$$

So, for example, if we take the formula $\varphi = (p \wedge q) \vee (\neg p \wedge \neg q)$, and the substitution $\mathbf{s} = \{(p \rightarrow r)/p\}$, we have:

$$\varphi\mathbf{s} = ((p \rightarrow r) \wedge q) \vee (\neg(p \rightarrow r) \wedge \neg q)$$

Observe that the substitution is applied only once. We do not recursively apply the substitution to the new formula, which would create an infinite loop of substitutions.

2.1.3 Normal Forms

At several points in this thesis, we shall rely on normal forms to simplify an argument. It is clear from the semantics and our use of derived operations that there are various formulas that have equivalent truth values. The idea behind a normal form is to use such equivalences to produce formulas of known structure. For instance, we have the De Morgan laws:

$$p \wedge q = \neg(\neg p \vee \neg q)$$
$$p \vee q = \neg(\neg p \wedge \neg q)$$

We could use the De Morgan laws to remove all instances of the $\wedge$ operator from a formula.

An example of normal form which we shall refer to later is negation normal form. Placing a formula into negation normal form is a two stage process. First convert all instances of the $\rightarrow$ operator into instances of the $\neg$ operator, using the conversion $\varphi \rightarrow \psi = \psi \vee \neg\varphi$. The second step of the conversion is to "push in" the $\neg$ operator so that in any subformula of the form $\neg\varphi$, φ is a single propositional variable. To do this, we use the De Morgan laws and eliminate double negation:

$$\neg(\varphi \vee \psi) = (\neg\varphi) \wedge (\neg\psi)$$
$$\neg(\varphi \wedge \psi) = (\neg\varphi) \vee (\neg\psi)$$
$$\neg\neg\varphi = \varphi$$

Repeated application of these three laws will produce a new formula that has the same truth value as the old formula. However, it will have no instances of $\rightarrow$, and all $\neg$ instances will be immediately followed by a propositional variable. For example, the formula $(p \wedge q) \rightarrow r$ becomes $r \vee \neg(p \wedge q)$, and then $r \vee (\neg p \vee \neg q)$ when converting to negation normal form.

When using normal forms, especially for automated reasoning, the complexity of the the process, and the complexity of the resulting formula are important considerations. We use normal forms to simplify our reasoning, but the complexity of reasoning about an arbitrary formula is based on the complexity of reasoning about the simpler formula, plus the complexity of reducing the arbitrary formula to normal form. If the conversion process is exponential in complexity, then it does not matter if we can apply a simpler reasoning to the resultant formula, reasoning about arbitrary formulas will require exponential complexity.

The negation normal form is not exponential in complexity. The conversion of $\varphi \rightarrow \psi$ into $\psi \vee \neg\varphi$ replaces one operator with two, for a linear change in complexity. Eliminating double negation actually reduces complexity, and, in the worst case, the De Morgan laws will take a formula of length l to one of length $2l$, a linear multiplication of the complexity of a formula. In the extreme case, a single $\neg$ operator can be turned into a $\neg$ operator in front of every single propositional variable in a formula. For example:

$$\neg(p \vee (q \wedge (r \vee p))) = \neg p \wedge (\neg q \vee (\neg r \wedge \neg p))$$

2.2 Logics

A **logic L** is considered to be a set of formulas, closed under some set of inference rules. An extension of the logic by a formula φ, written $\mathbf{L} \bigoplus \varphi$ is the closure of the set $\mathbf{L} \bigcup \{\varphi\}$ under the same rules. We call the basic set of formulas, before taking the closure, the **axioms** of the logic. We call any member of the set **L** a **theorem** of the logic **L**.

2.2.1 Classical Logic

For an example, take classical logic, the simplest logic that will be of interest within this thesis. One set of axioms for classical logic, taken from Chagrov and Zakharyaschev [10] is:

A1	$p_0 \rightarrow (p_1 \rightarrow p_0)$
A2	$(p_0 \rightarrow (p_1 \rightarrow p_2)) \rightarrow ((p_0 \rightarrow p_1) \rightarrow (p_0 \rightarrow p_2))$
A3	$p_0 \wedge p_1 \rightarrow p_0$
A4	$p_0 \wedge p_1 \rightarrow p_1$
A5	$p_0 \rightarrow (p_1 \rightarrow p_0 \wedge p_1)$
A6	$p_0 \rightarrow p_0 \vee p_1$
A7	$p_1 \rightarrow p_0 \vee p_1$
A8	$(p_0 \rightarrow p_2) \rightarrow ((p_1 \rightarrow p_2) \rightarrow (p_0 \vee p_1 \rightarrow p_2))$
A9	$\bot \rightarrow p_0$
A10	$p_0 \vee (p_0 \rightarrow \bot)$

Classical logic further has these two rules of inference:

Modus Ponens: From φ and $\varphi \rightarrow \psi$, we can infer ψ.

Substitution: From a formula φ, we can infer $\varphi\mathbf{s}$ for any substitution **s**.

A **derivation** of a formula φ is a finite sequence of formulas $\psi_1 \ldots \psi_n = \varphi$, where each ψ_i is either an axiom, created by applying the substitution to some earlier ψ_j, or created by applying the rule Modus Ponens to some earlier elements of the sequence ψ_j, ψ_k. Clearly, to make the rule of Modus Ponens applicable, ψ_k must have the form $\psi_j \rightarrow \psi_i$. If such a derivation exists for a formula, we say it can be derived.

The logic defined by the calculus consists of the set of all formulas that can be derived. A calculus such as this is said to be sound if every formula that can

be derived in this manner is a semantically valid formula. It is complete if every formula that is semantically valid can be derived.

This calculus is a sound and complete with respect to the semantics defined in Section 2.1.1 [10].

2.3 Modal Logic

Modal logic extends our syntax by adding a new unary operator, $\Box$. We may define its dual operator, $\Diamond$ by $\Diamond\varphi = \neg\Box\neg\varphi$. This $\Box$ operator can have many interpretations, however, one common interpretation is that $\Box$ denotes **necessity**.[1] Under this interpretation, $\Diamond\varphi$ then denotes that that "not φ is not necessary". That is, φ is **possible**.

2.3.1 Syntax of K

The basic normal modal logic is **K**. It starts with the same axioms and inference rules as classical logic, and then adds the axioms and inference rules for a new operator $\Box$. Thus, our method for formula construction becomes:

$\bot$ is a formula, as is any propositional variable p.

If φ and ψ are formulas, then so are:

$$\varphi \wedge \psi \quad \varphi \wedge \psi \quad \varphi \rightarrow \psi \quad (\varphi) \quad \Box\varphi.$$

Compared to the definition in Section 2.1, the only addition is the clause that if φ is a formula, so is $\Box\varphi$.

For our modal calculus, this addition is meaningless without some way of introducing formulas of the form $\Box\varphi$ into derivations. Thus, we add a new axiom to the logic:

A11 $\Box(p_0 \rightarrow p_1) \rightarrow (\Box p_0 \rightarrow \Box p_1)$

and a new inference rule:

Rule of Necessitation: From φ, we can infer $\Box\varphi$.

Just as the classical calculus was sound and complete with respect to the semantics outlined in Section 2.1.1, this new calculus is sound and complete with respect to the semantics of **K**, outlined formally at the end of Section 2.3.2 [10].

[1] Other interpretations include provability ($\Box\varphi$ means φ can be proven), tense ($\Box\varphi$ means φ is true in all future states) and epistemic ($\Box\varphi$ means φ is known to be true).

2.3.2 Semantics of K

The semantics of **K** hinge on the interpretation of $\Box$. Interpreting it as a necessity operator, we use the semantics of **possible worlds**. In classical logic, a statement is either true or false. However, in reality, some statements can be circumstantially true.

We formalise our intuition of modal logic using Kripke frames. Kripke semantics is a formal semantics for modal logic. The basic element of this semantics is the **Kripke frame**. A Kripke frame lays out some set of possible worlds, and an accessibility relation between the worlds. A Kripke **model** adds a valuation. Where classical valuations map propositional variables to true or false values, a valuation in a Kripke model maps world-variable pairs to true/false values.

Within a world of a Kripke model, a valuation defines the values that propositional variables have at that world. Formulas without any occurrence of the modal operators are evaluated at that world just as they would be in classical logic.

If a formula has the form $\Box\varphi$, then it is true at a world if φ is true at all worlds related to that world under our accessibility relation. Dually, a formula of the form $\Diamond\varphi$ is true at a world if φ is true at *some* world related to that world under the accessibility relation.

Some formulas will be true at every world in a model. We say these formulas are true in the model. Some formulas will be true in every model based on a particular frame. We say these formulas are true in that frame.

When a formula φ is not true at some world in a Kripke model, we say that the model is a **countermodel** for φ.

Formally, a Kripke frame is a pair $\langle W, R\rangle$, where W is a set of worlds and $R \subseteq W \times W$ a relation between worlds. A valuation V on a Kripke frame is a map from propositional variables to sets of worlds. If $w \in V(p)$, then p is true at the world w. A Kripke model is the triple $\langle W, R, V\rangle$.

When we need to refer to particular Kripke frames, we generally use a particular font. We denote frames with letters $\mathcal{F}, \mathcal{G}$, and models with letters $\mathcal{M}, \mathcal{N}$.

As with Section 2.1.1, we build the truth of a formula at a given world in a model inductively:

Given a Kripke model $\langle W, R, V\rangle$, a world $w \in W$, we construct a function f_w to map formulas to truth values at the world w:

$$f_w(\bot) = \text{false}$$

$$f_w(p) = \begin{cases} \text{true} & \text{if } w \in V(p) \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_w(\varphi \wedge \psi) = \begin{cases} \text{true} & \text{if } f_w(\varphi) = \text{true and } f_w(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_w(\varphi \vee \psi) = \begin{cases} \text{true} & \text{if } f_w(\varphi) = \text{true or } f_w(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_w(\varphi \rightarrow \psi) = \begin{cases} \text{true} & \text{if } f_w(\varphi) = \text{false or } f_w(\psi) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

$$f_w(\Box\varphi) = \begin{cases} \text{true} & \text{if } \forall x \in W, \text{ if } wRx \text{ then } f_x(\varphi) = \text{true} \\ \text{false} & \text{if } \exists x \in W \text{ such that } wRx \text{ and } f_x(\varphi) = \text{false} \end{cases}$$

A model $\mathcal{M} = \langle W, R, V \rangle$ validates a formula φ, or equivalently, φ is true in the model if, for all $w \in W$, $f_w(\varphi) = \text{true}$. We write this $\mathcal{M} \vDash \varphi$.

A frame $\mathcal{F} = \langle W, R \rangle$ validates a formula φ, or equivalently, φ is true in the frame, if for all possible valuations V, the model $\langle W, R, V \rangle$ validates φ. We write this $\mathcal{F} \vDash \varphi$.

A frame $\mathcal{F}$ validates a logic $\mathbf{L}$ if $\mathcal{F} \vDash \varphi$ for all formulas $\varphi \in \mathbf{L}$. We write this $\mathcal{F} \vDash \mathbf{L}$. A frame $\langle W, R \rangle$ is finite if the set W is finite.

A logic is **Kripke complete** if there exists a set of frames S such that for every $F \in S$, $F \vDash \mathbf{L}$, and for every formula $\varphi \notin \mathbf{L}$, there exists a frame $F \in S$ such that $\mathcal{F} \nvDash \varphi$. A logic has the **Finite Model Property** (FMP) if it is Kripke complete with respect to a set of finite frames.

All logics with the finite model property are of course Kripke complete, but not all Kripke complete logics have the finite model property, and not all logics are Kripke complete. Examples of logics without the FMP can be found in [17], while some logics without Kripke completeness can be found in [15]. Fortunately, this thesis deals in logics that are Kripke complete, so the issues of representing non-Kripke complete logics need not arise.

The modal logic $\mathbf{K}$ is the set of formulas that is validated by every possible Kripke frame. That is, for a formula $\varphi \in \mathbf{K}$, and an arbitrary Kripke frame $\mathcal{F}$, $\mathcal{F} \vDash \varphi$.

2.3.3 Logics KTB and S4

Two extensions of the logic **K** are **KTB** and **S4**. Both add axioms that can easily be defined as restrictions on frames. They are both normal extensions of the logic **K**. That is, both **KTB** and **S4** are closed under the operations of Modus Ponens, Substitution and Necessitation[2]

The first, **KTB**, adds two axioms to the axiom set of **K**. These axioms are $\Box p \rightarrow p$ and $\Diamond\Box p \rightarrow p$. We write this as $\mathbf{KTB} = \mathbf{K} \bigoplus \Box p \rightarrow p \bigoplus \Diamond\Box p \rightarrow p$, where the $\bigoplus$ operator indicates adding an axiom and taking the closure under the three inference rules of **K**. The axiom $\Box p \rightarrow p$ is the axiom of reflexivity. A frame $\langle W, R\rangle$ validates $\Box p \rightarrow p$ iff for all $w \in W$, it is the case that wRw. The axiom $\Diamond\Box p \rightarrow p$ is the axiom of symmetry. A frame $\langle W, R\rangle$ validates $\Diamond\Box p \rightarrow p$ iff for all $w, v \in W$, it is the case that wRv implies vRw.

The logic **S4** also adds two axioms to **K**. As before, we add $\Box p \rightarrow p$, the reflexivity axiom. The second axiom we add is $\Box p \rightarrow \Box\Box p$, the axiom of transitivity. A frame $\langle W, R\rangle$ validates $\Box p \rightarrow \Box\Box p$ iff for all $w, v, u \in W$, it is the case that wRv and vRu, together, imply that wRu.

An important consideration for axioms is that as $\Box$ and $\Diamond$ are dual operators, every axiom involving one modal operator has an equivalent using the other. Thus, reflexivity, $\Box p \rightarrow p$, may also be written $p \rightarrow \Diamond p$. Likewise, symmetry can be written $p \rightarrow \Box\Diamond p$ and transitivity may be written $\Diamond\Diamond p \rightarrow p$.

2.4 Sequent Calculi

An alternative formalism for discussing logics is the use of sequent calculi, also known as Gentzen systems. Where our prior formalism involved many axioms and few inference rules, a sequent calculus involves only a couple of axioms, and many inference rules.

The strength of sequent calculi that concerns us in this thesis is their applicability to automated reasoning. Unlike the semantics of the previous section, the inference rules of the sequent calculus can be applied backward with ease. For example, given a formula $\varphi \wedge \psi$, we can use an inference rule to state that this formula only belongs to our logic if both φ and ψ belong to our logic.

Thus, repeated backwards application of our inference rules allows us to work backwards, going from one formula to several simpler formulas, until we eventually reach formulas that are axiomatically true. Of course, it is also possible to

[2]There exist subnormal logics, but these are outside the scope of this thesis.

attempt to derive a formula that is not true. In this case, our derivation will reach a state where there is no applicable inference rule, but none of the formulas we have reached are axiomatically true, in which case we terminate the backwards application, declaring that this formula is not a formula of our logic.[3]

For the sake of simplicity, this thesis shall consider only single-sided sequent calculi, without explicit structural rules. The traditional structural rules of weakening (adding extra formulas to a formula set), contraction (removing duplicate formulas from a formula set and permutation (reordering the formulas within a set) are still valid. They are simply not made explicit. Substructural logics omit one or more of these rules, and are not covered within this thesis.

All our sequent calculi shall be cut-free. That is, they do not contain the rule of inference known as cut. The cut rule states that if we can derive a formula φ from a formula set Γ, and we can derive a formula set Δ from φ, then we can derive a Δ from Γ. Including the cut rule can make derivations significantly shorter. However, for the calculi we discuss, it is not *necessary* for any derivation, and our primary use of sequent calculi within this thesis is for automated reasoning. The cut rule creates significant difficulties for automated processes.

Other kinds of sequent calculi exist, but are outside the scope of this thesis. Further, we shall assume all our formulas are in a modal variant of the negation normal form in Section 2.1.3. We describe the necessary modifications to handle modalities below. We take our discussion from Troelstra & Schwichtenberg [42].

2.4.1 Modal Negation Normal Form

As in the non-modal case, the first step in converting a formula to negation normal form is to convert all instances of the $\rightarrow$ operator into instances of the $\neg$ operator, using the conversion $\varphi \rightarrow \psi = \psi \vee \neg\varphi$.

The second step, pushing the $\neg$ in to simplify subformulas of the form $\neg\varphi$, requires an additional pair of rules to properly handle the modal $\Box$ and $\Diamond$ oper-

[3] Alternatively, as discussed later, backwards application may enter an infinite loop, never terminating. That's bad, we shall discuss ways of preventing this from happening.

ations. Thus, our new set of rules is:

$$\neg(\varphi \vee \psi) = (\neg\varphi) \wedge (\neg\psi)$$
$$\neg(\varphi \wedge \psi) = (\neg\varphi) \vee (\neg\psi)$$
$$\neg\Box\varphi = \Diamond\neg\varphi$$
$$\neg\Diamond\varphi = \Box\neg\varphi$$
$$\neg\neg\varphi = \varphi$$

Again, repeated application of these equalities will eventually reach a formula where all instances of $\neg$ are immediately followed by a propositional variable. As before, converting a formula to negation normal form is a non-exponential process. The only new rules handle modalities, and these rules do not actually change the length of the formula.

2.4.2 Basic sequent calculus

We shall begin by laying out a sequent calculus for classical logic, without any modal operators. A sequent in our one-sided calculus is a set of formulas. We say a sequent $\varphi_1, \ldots \varphi_n$ holds if the disjunction of all the formulas $\varphi_1 \vee \ldots \vee \varphi_n$ is true. When we need to discuss individual elements of the sequent, we can consider the comma to act as a form of set union. The sequent written Γ, φ, Δ is the formula set $\Gamma \cup \{\varphi\} \cup \Delta$.

In classical logic, we have the axiom A10: $p_0 \vee (p_0 \rightarrow \bot)$. We shall use this as the basis for the axiom of our calculus. A sequent $\varphi_1, \ldots \varphi_n$ holds axiomatically if there exist some i, j such that $\varphi_i = p$ and $\varphi_j = \neg p$ for some propositional variable p.

For our inference rules, we need to handle the various operators of our logic. Since we have reduced our formulas to negation normal form, we only need to handle the operations of $\wedge$ and $\vee$.

The operation of $\vee$ is simple. As we said, a sequent $\varphi_1, \ldots \varphi_n$ holds if the disjunction of all its the formulas $\varphi_1 \vee \ldots \vee \varphi_n$ is true. As such, for any set of formulas Γ, the sequent Γ, φ, ψ holds if and only if the sequent $\Gamma, \varphi \vee \psi$ holds.

The second inference rule, for $\wedge$, requires slightly more complexity. Recalling our semantics for classical logic, we observe that if both the sequents Γ, φ and Γ, ψ hold, then so will $\Gamma, \varphi \wedge \psi$. Thus, unlike our $\vee$ rule, the rule for $\wedge$ will require two sequents for premises.

We write these rules in Figure 2.1. For the rules ($\wedge$) and ($\vee$), the sequent(s) above the line are the premise, and the sequent below the line is the conclusion.

$$(Axiom)\ \ p, \neg p, \Gamma \qquad (Verum)\ \ \top, \Gamma$$

$$(\vee)\ \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$$

$$(\wedge)\ \frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$$

Figure 2.1: Basic inference rules

The conclusion holds if all the sequents in the premise hold. The axioms are sequents that always hold, as discussed above.

Now, we can either read these rules forward, as taking axioms and building more complex formulas until we have the desired proof, or backwards, taking complex formulas and breaking them down into subformulas until we reach axioms. We call the set of applied rules that go from axioms to conclusions a proof tree. An example proof tree is in Figure 2.2, which derives $((p \wedge q) \vee \neg q) \vee \neg p$, which is the negation normal form of $p \rightarrow (q \rightarrow (p \wedge q))$.

$$\cfrac{\cfrac{\cfrac{\cfrac{\text{axiom}}{\neg p, \neg q, p}\ \cfrac{\text{axiom}}{\neg p, \neg q, q}}{\neg p, \neg q, p \wedge q}\,(\wedge)}{(p \wedge q) \vee \neg q, \neg p}\,(\vee)}{((p \wedge q) \vee \neg q) \vee \neg p}\,(\vee)$$

Figure 2.2: A simple prooftree

It is important that our sequent calculus reflects the semantics that we have previously outlined. As in Section 2.2.1, we care about the soundness and completeness of the logic. The inference rules in Figure 2.1 are sound and complete with respect to classical logic.

2.4.3 Modal Rules

The next important step for our calculus is to add rules to handle the addition of the $\Box$ and $\Diamond$ modalities to our formulas[4].

[4]Because we reduce everything to negation normal form, we cannot simply use the duality of $\Box$ and $\Diamond$ to handle both with one rule

Exactly what inference rules we shall need depend on our logic. The inference rules encode information that would previously have been placed in axioms. Thus, any modal logic we wish to encode will need its own inference rules.

We shall only present the inference rules for the logic **S4**. Firstly, this is the only logic we shall actually use with the sequent calculus. Secondly, for simplicity, we have chosen to present only cut-free sequent calculi. While cut-free formulations of **S4** are well known [14], there is no known cut-free calculus for **KTB**, the other main logic of interest. The modal logic **S5**, which has transitivity, reflexivity and symmetry does have cut-free formulations, but all require extra notation.

To create a sequent calculus for **S4**, we need to add two new rules to the rules in Figure 2.1. One shall handle $\Diamond$, and will embody reflexivity. The second shall handle $\Box$ formulas, and will allow us to properly handle transitivity. These two new rules are given in Figure 2.3.

$$(\Diamond)\ \frac{\Diamond\varphi, \varphi, \Gamma}{\Diamond\varphi, \Gamma} \qquad\qquad (\Box, \mathit{JUMP})\ \frac{\varphi, \Diamond\Delta}{\Box\varphi, \Diamond\Delta, \Gamma}$$

Figure 2.3: Modal rules for **S4**

It can be shown that using the rules in Figures 2.3 and 2.1, we create a calculus for **S4** that is both complete (can derive all formulas within **S4**) and sound (will not derive any formula not in **S4**) ([42], [23]).

To demonstrate the application of these modal rules, we shall derive the axioms $\Box p \rightarrow p$ and $\Box p \rightarrow \Box\Box p$ within our logic. First, we reduce the axioms to negation normal form. $\Box p \rightarrow p$ becomes $p \vee \Diamond\neg p$, and $\Box p \rightarrow \Box\Box p$ becomes $\Box\Box p \vee \Diamond\neg p$. Then, we may derive them as shown in Figure 2.4.

$$\cfrac{\cfrac{\cfrac{}{p, \Diamond\neg p, \neg p}\,\text{Axiom}}{p, \Diamond\neg p}\,(\Diamond)}{p \vee \Diamond\neg p}\,(\vee) \qquad\qquad \cfrac{\cfrac{\cfrac{\cfrac{}{p, \Diamond\neg p, \neg p}\,\text{Axiom}}{p, \Diamond\neg p}\,(\Diamond)}{\Box p, \Diamond\neg p}\,(\Box, \mathit{JUMP})}{\Box\Box p \vee \Diamond\neg p}\,(\vee)$$

Figure 2.4: Deriving the axioms of **S4** within our sequent calculus

One issue with these rules that is immediately apparent is that the basic inference rules for classical logic move from simple premises to more complex

conclusions. However, this is not necessarily the case with the modal rules. Inspection shows ($\Diamond$) has a premise that is no simpler than the conclusion. The ($\Box$, *JUMP*) rule can also cause problems. While human mediated derivations can deal with such things, when these modal rules are given to a computer for completely automated deduction, they can cause termination issues. These problems, and the ways they can be solved, are discussed in more depth in Chapter 5.

2.4.4 Creating countermodels

As mentioned before, it is possible to read the inference rules backwards, breaking a formula down into simpler rules. While we have mostly focused on successful proofs with our sequent calculus, it is possible to try and apply a backwards proof search to a formula that is not actually in the logic. In this case, if our search terminates, we shall eventually fail, and some branch of our proof tree will terminate in a sequent to which our inference rules do not apply. Using the sequent calculus provided above, the non-termination of our search is a distinct possibility.

However, if the proof search does terminate in a failure, it is actually possible to use our failed attempt at a proof to produce a countermodel for the formula under consideration. That is, we view the backwards application of rules not as creating a derivation of the formula from axioms, but as an attempt to create a countermodel. In this case, the semantic meaning of our rules changes. Instead of having a sequent to be proven, the sequent contains the set of formulas we are attempting to render false. If our derivation reaches the axiom $p, \neg p$, this means that we would simultaneously need p and $\neg p$ to be false to create a countermodel (which is clearly impossible).

This is the origin of the name ($\Box$, *JUMP*) for our $\Box$ rule; To create a model in which the formula $\Box\varphi$ in the conclusion of the sequent is false, we must move (jump) to a new world, the premise, and demonstrate that we can make the formula φ false. And of course, if we are to falsify a formula of the form $\Diamond\psi$, it must also be false in the new world, so the rule ($\Box$, *JUMP*) preserves formulas of the form $\Diamond\psi$, even though it discards most other formulas when read backwards.

Examples of this process can be found in a number of sources [2], [19]. In this thesis, we shall not need to do this. Our focus will be on termination of the **S4** calculus when backwards reasoning is applied. The creation of models from the backwards reasoning process is not something we consider.

One issue that arises when performing a backwards proof search is the im-

portance of backtracking. Read backwards, a rule like $(\Box, \mathit{JUMP})$ states that when given formulas of the form $\Box\varphi$, we choose to focus on a single formula and discard the other $\Box$ formulas. However, the process of countermodel construction can only work if all the formulas of the sequent are falsified. Thus, if our backwards reasoning does not terminate in the axiom, we must go back to the sequent where we applied the rule $(\Box, \mathit{JUMP})$ and check for other possible rule applications (including applying $(\Box, \mathit{JUMP})$ to other formulas of the form $\Box\varphi$).

Consider, for example, a sequent of the form $\Diamond\Box p, \Box\top, \Box p$. If we are reasoning backward, and apply the $(\Box, \mathit{JUMP})$ rule to the formula $\Box p$, we get the sequent $\Diamond\Box p, p$. Here, by repeated application of the rules $(\Diamond)$ and $(\Box, \mathit{JUMP})$, we could loop forever. On the other hand, if we apply the $(\Box, \mathit{JUMP})$ rule to the formula $\Box\top$, then we get the sequent $\Diamond\Box p, \top$, and our axiom (Verum) is immediately applicable, terminating our attempt with a successful derivation.

2.5 Algebra

Algebras form the third tool that we shall use to discuss logics. While there exist Kripke incomplete logics, algebras are a more general structure, capable of representing arbitrary logics without completeness issues. This comes at a cost, as algebras are also more complex and less intuitive than Kripke frames.

Universal algebra is an immense field, and this introduction is the bare minimum needed to understand the algebras used within this thesis. Readers interested in more detail are urged to seek out one of the numerous textbooks on the subject. Burris & Sankappanavar [8] provide an explanation of Universal Algebra, the Handbook of Modal Logic [7] explains the connection between logic and algebra in more detail and the three volume Handbook of Boolean Algebras [33] gives substantially more detail on boolean algebras. This introduction draws on all three of these sources, and others, as needed.

2.5.1 Boolean Algebras

Thanks to the well known Stone Isomorphism theorem [39], all boolean algebras are in fact isomorphic to algebras of sets. Here, we explain this treatment and how it relates to traditional classical logic.

To begin, we define a boolean algebra as a set A, containing two distinguished elements which we name 0 and 1, with some operators on the set: Two binary

operators $+$ and $\cdot$, and a unary operator $-$. We frequently identify the set A and the associated algebra, writing $\mathbf{A}$ for the tuple $\langle A, +, \cdot, -, 0, 1\rangle$.

To be a boolean algebra, the operators must obey some basic properties. We take the following list of properties from the Handbook of Boolean Algebras [33]:

Associativity	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Commutativity	$x + y = y + x$	$x \cdot y = y \cdot x$
Absorption	$x + (x \cdot y) = x$	$x \cdot (x + y) = x$
Distributivity	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
Complementation	$x + (-x) = 1$	$x \cdot (-x) = 0$

All Boolean algebras have a canonical partial order $\leq$, defined by $x \leq y$ iff $x + y = y$ (or, equivalently, $x \leq y$ iff $x \cdot y = x$). Further, under this canonical order, 0 is the least element of the algebra and 1 is the greatest. Of course, from $\leq$ we may define $<$, $>$ and similar operations in the usual manner.

One of the simplest examples of a Boolean algebra is the power set algebra. Given an arbitrary set[5] X, we define a boolean algebra on its power set $P(X)$, where 0 is the empty set, 1 is X, $+$ is the operation $\cup$, $\cdot$ is the operation $\cap$ and $-A$ is the complement $X \backslash A$ of A with respect to X.

The canonical partial order $\leq$ on a power set algebra is simply the order defined by subset inclusion.

Now, for an algebra, we can define **subalgebras**. For an algebra $\mathbf{B}$, defined on a set B to be a subalgebra of an algebra $\mathbf{A}$, defined on a set A, B must be a subset of A, and the operators of $\mathbf{B}$ must be the operators of $\mathbf{A}$, restricted to operating on the set B.

A **homomorphism** is a structure preserving map. Given two algebras $\mathbf{A} = \langle A, +_A, \cdot_A, -_A, 0_A, 1_A\rangle$ and $\mathbf{B} = \langle B, +_B, \cdot_B, -_B, 0_B, 1_B\rangle$, we say that a function $f : A \to B$ is a homomorphism if the operations are preserved. That is, for all x, y in A:

1. $f(0_A) = 0_B$, $f(1_A) = 1_B$
2. $f(x +_A y) = f(x) +_B f(y)$, $f(x \cdot_A y) = f(x) \cdot_B f(y)$
3. $f(-_A x) = -_B f(x)$

The function f is an **isomorphism** if the function is bijective. If such a function exists, we say $\mathbf{A}$ and $\mathbf{B}$ are isomorphic.

[5]This set X is not necessarily finite. If it were, many of our definitions would become significantly simpler.

An **algebra of sets** is either a power set algebra, or some subalgebra of a power set algebra. While it is possible to create Boolean algebras in other ways, in 1936, Stone [39] proved a representation theorem. We provide the aspect of it most significant to our work:

Theorem 2.1 (Stone's representation theorem). *All boolean algebras are isomorphic to some algebra of sets.*

As such, we may assume when dealing with a Boolean algebra that it is an algebra of sets.

Important Terms

When discussing algebras, some terms will come up repeatedly. Here we define the various terms we shall use. These terms are not defined in alphabetical order, but rather so a new reader may read from start to finish and understand each new definition by reference to prior definitions.

Many of these terms are used in universal algebra, which requires somewhat more general definitions. Here we only concern ourselves with the applications and definitions that this thesis will apply to Boolean algebras.

Lattice A lattice is a partially ordered set in which any two elements have a unique least upper bound and a unique greatest lower bound. All Boolean algebras are lattices under the canonical order $\leq$.

Lattices have naturally defined meet ($\wedge$) and join ($\vee$) operations. The meet of two elements is their unique greatest lower bound. The join of two elements is their unique least lower bound. A lattice is **distributive** if these two operations distribute over each other. That is, $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ and $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.

All boolean algebras are distributive, using $\cdot$ for the meet operation and $+$ for the join operation.

Atom An element a of an algebra is an atom if there is no other element x such that $0 < x < a$. In a power set algebra $\mathbf{P(X)}$, the atoms are singleton sets $\{x\}$, for $x \in X$. An algebra is **atomic** if for each element x of the algebra (except 0), there exists an atom $a \leq x$. All finite algebras are atomic.

Term A term is a function, defined using our existing operators and variables. x, $x \cdot y$, $x + y \cdot 1$ are all terms of Boolean algebra, and any element of the algebra may be used in place of the variables x and y to produce a value.

Filter For a Boolean algebra $\mathbf{A}$, $p \subseteq A$ is a filter if:

- $1 \in p$ and,
- $\forall x \in p$ and $\forall y \in A, x \leq y$ implies $y \in p$ and
- $\forall x, y \in p$, $x \cdot y \in p$.

If $0 \in p$, then $p = A$. For all elements $x \in A$, the set $p = \{y : x \leq y\}$ is a filter. We call this the **principal filter generated by** x. It is the **trivial filter** if $x = 1$. If $0 \notin p$, we say p is a **proper filter**.

Ultrafilter A filter p of an algebra $\mathbf{A}$ is an ultrafilter if for each $x \in A$, it is the case that either $x \in p$ or $-x \in p$, but not both.

Some other properties are equivalent to being an ultrafilter. A filter p is **prime** if it is a proper filter and $\forall x, y \in A$, $x + y \in p$ implies $x \in p$ or $y \in p$. A filter p is **maximal** if it is a proper filter and there is no proper filter q such that p is a proper subset of q.

The three properties of being prime, maximal and an ultrafilter are equivalent. The principal filter generated by an arbitrary element a is an ultrafilter iff a is an atom of $\mathbf{A}$. In a finite algebra, all ultrafilters are principal filters.

Congruence For an algebra $\mathbf{A}$, an equivalence relation $\sim$ on A is an congruence if for all x, x', y, y' in A, $x \sim x'$ and $y \sim y'$ implies $-x \sim -x'$ and $x + y \sim x' + y'$. That is, the equivalence relation respects the operations of $-$ and $+$ (and so, by duality, also respects the operation of $\cdot$).

All algebras have two trivial congruences. The first comes from the universal equivalence relation, where all elements of the algebra are considered equivalent. The second comes from the identity relation, where no element of the algebra is considered equivalent to any element except itself.

If an equivalence relation $\sim$ is a congruence, then the relation $\sim$ is entirely defined by the set of elements x such that $x \sim 1$. A congruence $\sim$ is finer than a congruence θ if $x \sim y$ implies $x \theta y$. Using this as an ordering, the set of all possible congruences on an algebra form a lattice.

We say an algebra is **Congruence Distributive** if the lattice of all possible congruences on the algebra is distributive. Boolean algebras are congruence distributive.

Congruence Extension Property An algebra $\mathbf{A}$ has the congruence extension property if for every subalgebra $\mathbf{B}$ of $\mathbf{A}$, and every congruence $\sim$ on $\mathbf{B}$, there is a congruence θ on $\mathbf{A}$ such that $\sim = \theta \cap B^2$.

Quotient Algebra Given a congruence $\sim$ on $\mathbf{A}$, the quotient algebra of $\mathbf{A}$ by $\sim$, written $\mathbf{A}/\sim$ is the algebra based on the set of equivalence classes defined by $\sim$, and with operations defined in a natural way: $a/\sim + b/\sim = (a+b)/\sim$.

Direct Product The direct product of two algebras $\mathbf{A}$ and $\mathbf{B}$ is defined by taking the Cartesian product $A \times B$, and defining all operations componentwise. That is, $(a_1, b_1) + (a_2, b_2) = (a_1 +_{\mathbf{A}} a_2, b_1 +_{\mathbf{B}} b_2)$. This definition can be extended to the product of more than 2 algebras in the obvious way.

Further, we can take the product of a set of algebras $\Pi_{i \in I} \mathbf{A_i}$, where I is an arbitrary set. In particular, the set I can be part of some other algebra $\mathbf{I}$. In this case, an ultrafilter U in the algebra $\mathbf{I}$ will define a congruence on $\Pi_{i \in I} \mathbf{A_i}$, by $a \sim b$ if $\forall i \in U, a_i = b_i$.

Ultraproduct An ultraproduct is defined with an index algebra $\mathbf{I}$, a set of algebras A_i, for $i \in I$, and an ultrafilter U in $\mathbf{I}$. As mentioned in the definition of product, the ultrafilter U defines a congruence $\sim$ in the product algebra $\Pi_{i \in I} A_i$. The ultraproduct is the quotient algebra created by applying this congruence to the product algebra. We write this $\Pi_{i \in I} A_i/\sim$.

Note that if I is a finite set, this definition is uninteresting, since then the ultrafilter will be defined by an atom a of $\mathbf{I}$, and the quotient algebra will be isomorphic to the algebra A_a.

Discriminator A discriminator function t is a ternary operation defined by

$$t(a,b,c) = \begin{cases} a & \text{if } a \neq b \\ c & \text{if } a = b \end{cases}$$

An algebra has a discriminator term if there is some ternary term $t(x,y,z)$ in the algebra that represents the discriminator function. That is, if elements of the algebra a, b, c are used in place of the variables x, y, z, and the term evaluated, the result given is a if $a \neq b$, or c if $a = b$.

Simple Algebra A simple algebra is an algebra in which the only possible congruences are the trivial ones. All algebras with a discriminator term are simple.

Variety A class of algebras is a variety if it is closed under homomorphisms, subalgebras and direct products. That is, taking a homomorphism or subalgebra of a member of a variety, or taking the direct product of multiple members of the variety will produce a member of the variety. If K is a class of algebras, $V(K)$, the variety generated by K is the smallest variety containing K. Of course, K may contain only a single algebra $\mathbf{A}$, in which case we shall identify $\mathbf{A}$ and the class consisting of $\mathbf{A}$, and write $V(\mathbf{A})$.

Discriminator Variety If K is a class of algebras with a common discriminator term $t(x, y, z)$ then $V(K)$ is a discriminator variety.

Projection map Given a direct product of some family of algebras $\Pi_{i\in I}\mathbf{A}_i$, the projection map onto the jth coordinate $\pi_j : \Pi_{i\in I}\mathbf{A}_i \to A_j$ is defined by $\pi_j(\{a_i : i \in I\}) = a_j$. This map is always a homomorphism.

Subdirect Product An algebra $\mathbf{B}$ is the subdirect product of an indexed family $(\mathbf{A}_i)_{i\in I}$ if $\mathbf{B}$ is a subalgebra of $\Pi_{i\in I}\mathbf{A}_i$, such that for each $i \in I$, the projection map π_i is a surjective mapping.

Subdirectly Irreducible Algebra An algebra is subdirectly irreducible if it cannot be expressed as the subdirect product of other algebras. Formally, if $\mathbf{B}$ is subdirectly irreducible, and $\mathbf{B}$ is isomorphic to some subdirect product of a family $\mathbf{A}_i$, then there exists some i such that $\mathbf{B}$ is isomorphic to $\mathbf{A}_i$.

Important Theorems

In addition to the previously mentioned Stone isomorphism, there are some other results that will be needed, and so bear mentioning explicitly.

Theorem 2.2 (Tarski). *Let K be a class of algebras. The variety generated by K is created by taking the class of all homomorphic images of subalgebras of products of elements of K (Written $\mathbf{HSP}(K)$) [40].*

We define the variety generated by K as the smallest class containing K closed under homomorphisms, subalgebras and direct products. This theorem tells us that we can generate all the elements of the variety by taking these operations in a particular order.

Theorem 2.3. *Every algebra can be subdirectly decomposed into subdirectly irreducible algebras. As a corollary, every variety is generated by its subdirectly irreducible members [5].*

This theorem tells us that subdirectly irreducible algebras are to algebras as prime numbers are to integers. Each algebra has a subdirect decomposition, just as each integer has a prime decomposition.

Theorem 2.4 (Jónnsson's Lemma). *Let K be a class of algebras such that the all elements of the variety generated by K are congruence distributive. Then all subdirectly irreducible members of the $V(K)$ belong to the class of homomorphic images of subalgebras of ultraproducts of members of K (Written $\mathbf{HSP_U}(K)$) [25].*

This lemma, together with the prior theorem, gives us another way of generating a variety. Boolean algebras with operators are always congruence distributive. Further, if K is a finite set of finite algebras, $\mathbf{P_U}(K) = K$, which greatly simplifies the problem.

Theorem 2.5. *If $V(K)$ is a discriminator variety, then all subdirectly irreducible members of $V(K)$ are simple algebras [8].*

Discriminator varieties have a number of useful properties. However, this is the only one that we will need.

Classical Logic

When dealing with classical logic, the various operators of Boolean algebras have natural isomorphisms. The distinguished elements 1 and 0 map to $\top$ and $\bot$, $+$ and $\cdot$ become $\vee$ and $\wedge$, while $-$ becomes $\neg$. With these natural isomorphisms, we can translate formulas of our logic directly to terms in the algebra.

The relation $\leq$ is akin to $\rightarrow$, in that for an algebra, $x \leq y$ iff $x \cdot -y = 1$, and for a logic, $x \rightarrow y$ is true iff $x \vee \neg y$ is true. However, they are not directly equivalent, as $\leq$ is a relation between elements of the algebra, where $\rightarrow$ is an operator, used to construct formulas.

From this, an isomorphism between algebras and classical logic becomes simple. Consider the two element Boolean algebra, in which 1 and 0 are the only elements. Given a valuation V that assigns truth values to propositional variables, we can simply use these natural isomorphisms to directly compute the truth value of a formula.

2.5.2 Modal Logic and Boolean Algebras with Operators

Just as modal logic adds an extra operator $\Box$ and its dual $\Diamond$ to classical logic, the algebraic analog adds an extra operator f, which functions similarly to $\Diamond$

to Boolean algebras. It is important to bear in mind that the structure thus described is actually dual to that of a Kripke frame so described. We have a theory of duality, rather than isomorphism.

That is, where a Kripke frame has a set of worlds W and a relation R, and the $\Diamond$ operator looks forward along the relation R, the operator f looks backwards along the relation R. Fortunately, within this thesis, our main concern with algebras is their connection to **KTB**, where the relationships are all symmetrical.

To formalise this duality, consider a Kripke frame $\langle W, R\rangle$. Now, take the power set algebra on W. That is, elements of our algebra are sets of worlds in our frame. To define the modal operation f, analogous to $\Diamond$, we define $f(x) = \{y \in W | yRx\}$. Thus, if x represents the set of worlds where φ is true, $f(x)$ represents the set of worlds where $\Diamond\varphi$ is true.

Now, a valuation, as previously defined, maps propositional variables to sets of worlds. In an algebra of sets, each element is also a set of worlds, so a valuation maps propositional variables to elements of the algebra.

As with classical logic, it is possible to turn formulas into terms, and calculate the truth value of a term directly. Thus, given an arbitrary Kripke frame, we can define an algebra dual to the frame.

When we define logics by adding axioms, the algebraic equivalent is to impose conditions on the operator f. For the basic logic **K**, there are only two conditions on the operator f. The first property is a distributive property: $f(x + y) = f(x)+f(y)$. This corresponds to the distributive property of our modal operators, $\Diamond(a \vee b) = \Diamond a \vee \Diamond b$. The second property is that $f(0) = 0$. This corresponds to the logical formula $\Diamond\bot = \bot$.

However, not all algebras are dual to a Kripke frame. Some algebras have no dual Kripke frame. This is not the case for any finite algebra. All finite algebras are atomic, and by treating their atoms as the worlds of a frame, we can get a dual frame. On the other hand, if we start dealing with non-atomic (and hence infinite) algebras, then it is possible to create an algebra with no dual Kripke frame. The classes of algebras with no dual Kripke frames correspond to the logics that are not Kripke complete.

There are tools that can deal with Kripke incomplete logics, such as general frames. A **general frame** takes the $\langle W, R\rangle$ pair of a Kripke frame, and adds an additional set I, containing subsets of W. The members of I are called distinguished worlds. The set I must be closed under the traditional set operations of $\cup, \cap, -$, and also the operation $\Box$, defined here as $\Box A = \{x \in W : \forall y \in W, xRy \rightarrow y \in A\}$. While reasoning about models based on general frames, we

work with the same definitions as a Kripke model. However, we restrict the potential valuations by adding a rule that for any valuation V, and any propositional variable p, $V(p)$ must be a member of our set I.

There are three important operations on algebras that preserve truth. These are taking homomorphisms, taking subalgebras and taking direct products [10]. Recalling Theorem 2.2, this means that if a class of algebras all make certain formulas true, so will the variety generated by that class. Just as a logic actually corresponds to some set of Kripke frames, a logic corresponds to a variety of algebras.

Extensions of a logic correspond to smaller sets of Kripke frames, and these extensions form a lattice under the relation of inclusion. Likewise, extensions of a logic correspond to smaller varieties, and the extensions form a lattice under the relation of inclusion.

Modal Algebras for S4 and KTB

As mentioned, we can create modal algebras for specific logics by adding conditions to the possible values of the operator f. As the logics we look at are extensions of **K**, they inherit its restrictions of $f(0) = 0$ and $f(x + y) = f(x) + f(y)$.

For the modal logic **S4**, we add axioms of reflexivity and transitivity. Each of these corresponds to a restriction on f. Reflexivity, $\Box p \rightarrow p$, is dually $p \rightarrow \Diamond p$ and corresponds to a restriction that $x \leq f(x)$. Transitivity, $\Box p \rightarrow \Box\Box p$, is dually $\Diamond\Diamond p \rightarrow \Diamond p$ corresponds to a restriction that $f(f(x)) \leq f(x)$. If an atomic Boolean algebra has an operator with these two restrictions, as well as the restrictions inherited from **K**, then it is dual to a reflexive, transitive Kripke frame.

For the modal logic **KTB**, we add axioms of reflexivity and symmetry. Reflexivity is the same as for **S4**. For symmetry, $\Diamond\Box p \rightarrow p$, we add a restriction that $x \leq -f - f(x)$. Again, adding the restrictions for symmetry and reflexivity to the restrictions for **K**, we get a set of four restrictions. If the operator on an atomic Boolean algebra corresponds to these four restrictions, then it is dual to a reflexive, symmetric Kripke frame.

Chapter 3

Extensions of KTB

3.1 Introduction

The modal logic **KTB** is a normal extension of the modal logic **K**, explained in Chapter 2. The logic **KTB** adds to **K** a pair of axioms representing reflexivity and symmetry. This logic is Kripke complete with respect to the class of reflexive, symmetric frames. In a sense, **KTB** is a logic of graphs. Here, we consider extensions of **KTB**. Axiomatically, an extension involves taking the logic **KTB** and adding axioms. Semantically, it involves taking the set of undirected graphs and removing some to produce the logic of a smaller set of graphs.

If we consider the extensions of **KTB** to be logics with more axioms, then a "large" extension adds almost as many axioms as possible without causing a contradiction. Dually, if we consider an extension to be defined by a smaller set of graphs, then the large extensions are defined by very small sets of graphs. In this chapter, we will discuss the cardinality of these "large" extensions of **KTB**. We shall show that there exists a continuum of near-maximal extensions of **KTB**, where "near-maximal" means that the logic has at most one non-maximal extension.

3.2 Definitions

Throughout this chapter, we will be moving between Kripke frames, graphs and algebras. We shall establish a few conventions to make these transitions smoother.

Firstly, we shall use consistent fonts to denote logics $\mathbf{L}$, graphs A, B, C, Kripke frames $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and algebras $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$.

When using isomorphisms between our various kinds of structure, we shall

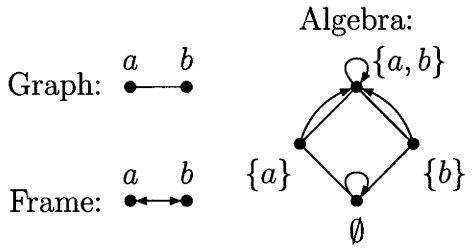

Figure 3.1: Equivalent structures K_2, $\mathcal{K}_2$ and $\mathfrak{K}_2$

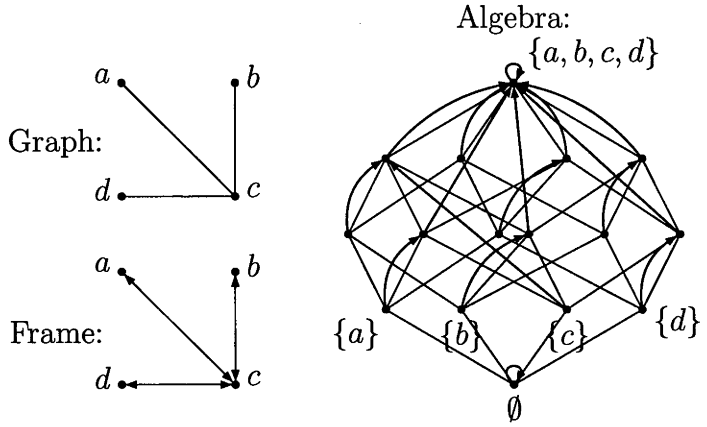

Figure 3.2: Equivalent structures; Note the rapid growth of the algebra

use consistent letters as we shift font; That is, if $G = (V, E)$ is a graph, then $\mathcal{G} = \langle V, R \rangle$ is a Kripke frame with the same universe and the reflexive closure of E for its accessibility relation R, and $\mathfrak{G}$ is the associated algebra.

Figure 3.1 shows this duality for a simple graph/frame/algebra set, while Figure 3.2 shows the duality for a more complex set. From inspection of these figures, it should be clear that, at least as far as diagrams are concerned, showing both the frame and graph is redundant, while showing the algebra rapidly gets too complex for any simple diagram.

As we progress through the chapter, we shall define several standard graphs. We begin with the complete graphs. K_i is the graph with i vertices and a universal accessibility relation, where every vertex is connected to every other vertex. $\mathcal{K}_i$ and $\mathfrak{K}_i$ are the associated Kripke frame and algebra, respectively.

We must define some commonly used terms for our various structures. Many of these definitions were in Chapter 2, but are repeated here for convenience.

A graph G is **connected** if for any pair of vertices a, b, there is a chain of

vertices $v_1 \dots v_n$ from V such that $a = v_1 E \dots E v_n = b$. The length of this chain is equal to the number of points, n, minus one. The **distance** between two vertices is the length of the shortest such chain. For purposes of this discussion, we will only be interested in connected graphs.

The **diameter** of a connected graph G is the greatest distance between any two points. That is, if a graph has diameter n, then any pair of points a, b will be connected by a chain of at most n points.

A **variety** of algebras is a class of algebras closed under homomorphisms, subalgebras and direct products.

An equivalence relation on an algebra is a **congruence** if it contains a subalgebra within a single equivalence class. A congruence is determined entirely by the set of elements that are in the same equivalence class as the identity element. Trivial congruences can be created by assuming that the identity element is the only member of its equivalence class, or by assuming every element is in the same class as the identity. An algebra is **simple** if the trivial congruences are the only congruences possible on the algebra.

A relation between graphs that we shall use later is that of the bounded morphism. A function $f : A \to B$ is a bounded morphism if it obeys the following two conditions:

Homomorphic Condition: For all elements $x, y \in A$, if xR_Ay then $f(x)R_Bf(y)$.

Back Condition: For all elements $x \in A$, and all elements $v \in B$, if $f(x)R_Bv$ then $\exists y \in A$ such that xR_Ay and that $f(y) = v$.

The concept of a bounded morphism extends naturally to frames and, for algebras, it has a natural connection to subalgebras[1]. If a frame F is a bounded morpic image of a frame G, then dually, $\mathfrak{F}$ is a subalgebra of $\mathfrak{G}$.

We define the modal logic **K** as outlined in Chapter 2.

The logic **KTB** adds the axioms $T = p \to \Diamond p$ and $B = p \to \Box\Diamond p$ to **K**. The axiom T is a reflexivity axiom; for a given frame $\mathcal{F}$, we have $\mathcal{F} \vDash T$ iff for every point x in the frame, xRx. The axiom B is a symmetry axiom; for a given frame $\mathcal{F}$, we have $\mathcal{F} \vDash B$, iff for every pair of points x, y in $\mathcal{F}, xRy \iff yRx$.

We shall be discussing the normal extensions of **KTB**, that is to say, the lattice NExt(**KTB**). A normal extension is created by adding a new formula φ

[1]Readers familiar with computer science may recall the concept of a bisimulation, which is a generalisation of bounded morphisms. Unlike a bisimulation, a bounded morphism is directional [6].

to the logic, and then taking the closure under the standard rules of deduction (Substitution, Modus Ponens and Necessitation), as outlined in Chapter 2.

There is a duality between general frames and boolean algebras, defined in Chapter 2. We can define a **KTB**-algebra as having the structure $\mathfrak{A} = \langle A; \wedge, \vee, \neg, f, 0, 1\rangle$, where $\langle A; \wedge, \vee, \neg, 0, 1\rangle$ is a Boolean algebra, and f is a unary operation satisfying the following conditions:

1. $f(0) = 0$
2. $f(x \vee y) = f(x) \vee f(y)$
3. $x \leq f(x)$
4. $x \leq \neg f(\neg f(x))$

The first two conditions are standard for a boolean algebra with operators. They are dual to the modal logic principles that $\Diamond\bot \leftrightarrow \bot$ and $\Diamond(p \vee q) \leftrightarrow \Diamond p \vee \Diamond q$. The third condition is the dual of reflexivity, $p \rightarrow \Diamond p$, and the fourth condition is dual to symmetry, $p \rightarrow \Box\Diamond p$. Both the third and fourth conditions can also be rendered as identities, using the standard rule that $x \leq y \Leftrightarrow x \wedge y = x \Leftrightarrow y \vee x = y$, so the class of **KTB**-algebras is a variety, which we shall write KTB.

We recall one of the manifestations of the duality between general frames and algebras:

Proposition 3.1. *If a graph G is connected and its diameter bounded by some positive integer k, then any modal algebra on $\mathcal{G}$ is simple and possesses a discriminator term.*

Note that if G is finite, its diameter *must* be bounded.

We shall denote by Λ(KTB) the lattice of subvarieties of KTB. Another manifestation of the duality of frames and algebras is:

Proposition 3.2. *The lattices NExt*(**KTB**) *and* Λ(KTB) *are dually isomorphic [26].*

3.3 Prior Results

The first result discovered about NExt(**KTB**) is that there is a single greatest extension, the logic defined by the frame $\mathcal{K}_1$. This comes from Makinson's Theorem [29] that all normal serial logics are sublogics of the identity logic.

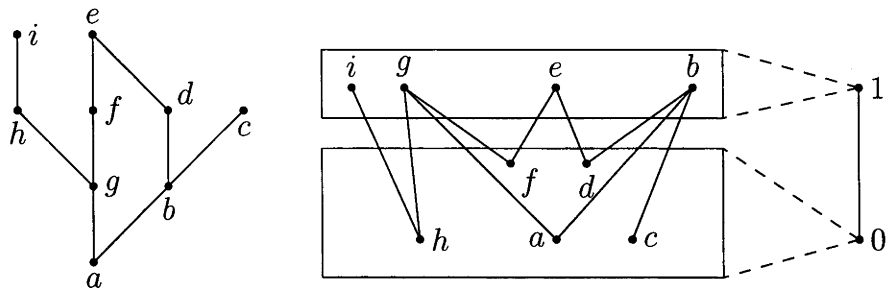

Figure 3.3: A simple frame, divided

The second result, harder to prove, comes from a theorem of Yutaka [31], and says that there is a unique second-greatest extension of **KTB**. This is the logic defined by the frame $\mathcal{K}_2$.

Yutaka proves that all frames $\mathcal{F}$ (other than $\mathcal{K}_1$) have a trivial bounded morphism to the frame $\mathcal{K}_2$. To create this bounded morphism, start with a single point x, and let $X = \{x\}$. Then take the set $Y = \{y : xRy\} - x$. Then define $X' = \{x : \exists y \in Y, yRx\} - Y$ and $Y' = \{y : \exists x \in X, xRy\} - X'$. Using X' and Y' as our new X and Y, repeat. This process can be iterated until it reaches a fixpoint.

Having reached the fixpoint, we create our bounded morphism mapping all elements of X to one point in the frame $\mathcal{K}_2$, and all elements of Y to the other. In Figure 3.3, we see an example of this, starting from the node a and dividing the frame into "upper" and "lower" sections.

Dually, these theorems give us that the lattice $\Lambda\mathsf{KTB}$ has a single atom $V(\mathfrak{K}_1)$, and this atom has a single cover $V(\mathfrak{K}_2)$. Here, we concern ourselves with the set of covers of $V(\mathfrak{K}_2)$.

3.4 The n-Spider

We now introduce the notion of n-spiders, which we use to demonstrate that there are at least a countable number of covers of $V(\mathfrak{K}_2)$ in the lattice $\Lambda(\mathsf{KTB})$. The n-spiders create a countable family $\{\mathfrak{S}_n\}_{n\in\omega}$ of finite **KTB**-algebras with the properties:

1. $\forall n \in \omega, \mathfrak{S}_n$ is a simple algebra

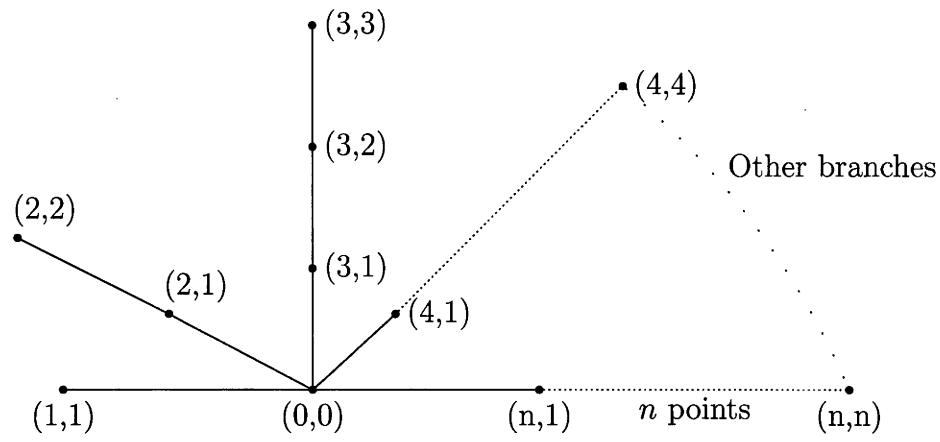

Figure 3.4: The n-spider graph

2. If $n \neq m$ then $\mathfrak{S}_n$ is not isomorphic to $\mathfrak{S}_m$

3. Each $\mathfrak{S}_n$ has exactly 2 proper subalgebras, $\mathfrak{K}_1$ and $\mathfrak{K}_2$.

Given a family of algebras with these properties, we can show:

Lemma 3.3. *For each $n \in \omega$, the variety $V(\mathfrak{S}_n)$ covers $V(\mathfrak{K}_2)$ in $\Lambda(\mathsf{KTB})$. Moreover, if $n \neq m$, then $V(\mathfrak{S}_n) \neq V(\mathfrak{S}_m)$.*

Proof. Let $\mathbb{V}$ be a subvariety of $V(\mathfrak{S}_n)$ for some $n \in \omega$. We may assume $\mathbb{V} = V(\mathfrak{A})$ for some subdirectly irreducible $\mathfrak{A}$.

By Jónsson's Lemma [25], $\mathfrak{A} \in \mathbf{HSP_U}(\mathfrak{S}_n)$. By the finiteness of $\mathfrak{S}_n$ and the Congruence Extension Property, we have $\mathbf{HSP_U}\mathfrak{S}_n = \mathbf{HS}(\mathfrak{S}_n) = \mathbf{SH}(\mathfrak{S}_n)$.

Because property 1 states $\mathfrak{S}_n$ is a simple algebra, $\mathbf{SH}(\mathfrak{S}_n) = \mathbf{S}(\{\mathfrak{S}_n, \mathfrak{T}\})$, where $\mathfrak{T}$ is the trivial algebra.

Property 3 gives us $S(\mathfrak{S}_n) = \{\mathfrak{K}_1, \mathfrak{K}_2, \mathfrak{S}_n\}$, and we have $\mathbf{S}(\mathfrak{S}_n) = V_{SI}(\mathfrak{S}_n)$, the subdirectly irreducible members of $V(\mathfrak{S}_n)$.

Therefore, $\mathbb{V}$ must be one of $V(\mathfrak{T}), V(\mathfrak{K}_1), V(\mathfrak{K}_2)$ or $V(\mathfrak{S}_n)$. Thus, $V(\mathfrak{S}_n)$ covers $V(\mathfrak{K}_2)$

If we assume that $V(\mathfrak{S}_n) = V(\mathfrak{S}_m)$, then we have $V_{SI}(\mathfrak{S}_n) = V_{SI}(\mathfrak{S}_m)$ and so $S(\mathfrak{S}_n) = S(\mathfrak{S}_m)$. From this, it follows that $\mathfrak{S}_n$ is isomorphic to $\mathfrak{S}_m$. ⊣

Thus, all we need to do is define the family $\{\mathfrak{S}_n\}_{n\in\omega}$. We do this through the duality of frames and algebras, and the isomorphism of graphs and frames.

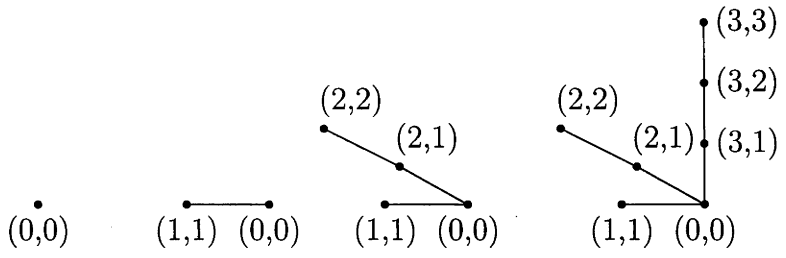

Figure 3.5: The graphs S_0, S_1, S_2, S_3

We define the graph $S_n = (V_n, E_n)$, shown in Figure 3.4. For the vertex set V_n, we take:

$V_n = \{(0,0), (1,1), (2,1), (2,2), \ldots, (m,1), \ldots, (m,m), \ldots, (n,1), \ldots, (n,n)\}$.

Then define a relation T_n, where $(p,k)T_n(q,l)$ iff either of the following conditions holds:

- $p = k = 0$ and $l = 1$
- $p = q$ and $l = k + 1$

Using this definition, T_n makes (V_n, T_n) a tree with root $(0,0)$ and branches $\langle (0,0), (k,1), \ldots, (k,k) \rangle$, for all $k \leq n$. For E_n, we take the reflexive, symmetric closure of T_n.

As shown in Figure 3.5, the first few graphs S_0, S_1, S_2 are all chains, while the graph S_3 begins to show signs of more interesting complexity. Figure 3.4 shows the general graph S_n.

Now, we prove that the family of algebras $\mathfrak{S}_n$, dual to our family of graphs, form a countable set of covers of $\mathfrak{K}_2$.

First, given a graph $G = (V, E)$, we recall that the valence of a vertex $x \in V$ is the number of outgoing edges. We write $\mathsf{val}(x)$ for the valence of x and $N(x)$ for the set $\{y \in V : yEx, y \neq x\}$. Thus, $\mathsf{val}(x) = |N(x)|$ for any $x \in G$.

Lemma 3.4. *Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ be graphs and $\mu : V_1 \to V_2$ a surjective bounded morphism. Then, $\mathsf{val}(x) \geq \mathsf{val}(\mu(x))$ for any $x \in V_1$.*

Proof. Let Q stand for $\{\mu(y) : y \in N(x)\}$. For any $x \in V_1$ we have $|\{\mu(y) : y \in N(x)\}| \leq |N(x)|$. Since μ is a bounded morphism, for every $y' \in N(\mu(x))$ there is a $y \in N(x)$ with $\mu(y) = y'$. Thus, $N(\mu(x)) \subseteq Q$, so $\mathsf{val}(\mu(x)) = |N(\mu(x))| \leq |Q| \leq |N(x)| = \mathsf{val}(x)$. ⊣

Lemma 3.5. *Let $G = (W, D)$ be a graph and $\mu : S_n \to G$ be a surjective bounded morphism. If there exists a distinct pair of points x and y in S_n such that $\mu(x) = \mu(y)$, then there exists a point $z \neq (0,0)$ in S_n such that $\mu(z) = \mu(0,0)$.*

Proof. Assume that there exists such a pair of points x, y, with $x = (m, i)$ and $y = (k, j)$. We must consider two cases, $i \neq j$ and $i = j$.

For the case $i \neq j$, assume, without loss of generality, that $i < j$, and perform an induction on the value of i. Firstly, if $i = 0$, our claim holds true trivially, with x being the point $(0,0)$, and y being our point z.

Suppose that the claim holds true for all $i' < i$. Then since $(m,i)E_n(m,i-1)$, we have $\mu((k,j)) = \mu((m,i))D\mu((m,i-1))$. So by the back condition, there exists a point a in S_n such that $\mu(a) = \mu((m,i-1))$ and $(k,j)E_n a$. Thus, $a \in \{(k,j-1),(k,j),(k,j+1)\}$. (If $k = j$, the point $(k,j+1)$ does not exist, and there are only 2 possibilities for a.)

Now, since $i < j$, we have $i-1 < j-1$, and so we can conclude that $a = (k,s)$, for some $s > i-1$, and the inductive hypothesis applies.

For the case $i = j$, assume, without loss of generality, $m < k$. If $i \neq m$, then there is a point $(m, i+1)$. Now, $(m,i)E_n(m,i+1) \Rightarrow \mu((m,i))D\mu((m,i+1))$. Thus, by the back condition, we have a point $a \in \{(k,j-1),(k,j),(k,j+1)\}$, such that $\mu(a) = \mu((m,i+1))$. If a is anything but the point $(k,j+1)$, then we reduce to the previous case of $i \neq j$.

Thus, we establish that if $i \neq m$, then either $\mu((m,i+1)) = \mu((k,j+1))$, or we have returned to the case of $i \neq j$. By an inductive process, this repeats until we have $\mu((m,m)) = \mu((k,m))$.

In this case, by Lemma 3.4, $\mu((m,m))$ has some unique point a such that $\mu((m,m))Da$. By the back condition, $a = \mu((m,m-1))$ Since $((k,m))E_n((k,m+1))$, we have $\mu((k,m))D\mu((k,m+1))$. Thus, $\mu((k,m+1))$ is equal to either $\mu((m,m))$ or $\mu((m,m-1))$. Either way, we have reduced this to the case of $i \neq j$. ⊣

Note that for small values of n, S_n is uninteresting; $S_0 = K_1$, $S_1 = K_2$, and S_2 is merely the 4 element chain. However, for larger values of n, we have the following lemma:

Lemma 3.6. *Let n be at least 2. If $\mu : S_n \to G$ is a surjective bounded morphism, then G is isomorphic to one of $\{K_1, K_2, S_n\}$.*

Proof. In this proof, let $S_n = (V_n, E_n)$, $G = (W, D)$.

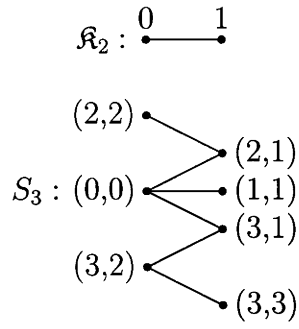

Figure 3.6: S_3 is the first interesting n-Spider. $\mathfrak{K}_2$ is its only nontrivial image

The lemma holds for S_2 in trivial fashion; The image of a chain must be a chain, and it is easy to see that the 3 element chain is not a bounded-morphic image of the 4 element chain. Figure 3.6 shows the morphism between S_3 and the two element chain. Note that the points $(2,2)$ and $(0,0)$ map to the same point, as do the points $(1,1)$ and $(3,3)$. Generalising this, along with an observation that bounded morphisms cannot *increase* the distance between two points, is pivotal to the following proof.

Assume that G is not isomorphic to K_1 or S_n. We show that it must be isomorphic to K_2.

Since G is not isomorphic to S_n, and μ is surjective, there exist a pair of points x and y in S_n such that $\mu(x) = \mu(y)$ and $x \neq y$. By Lemma 3.5, there must be some point a such that $a \neq (0,0)$, but $\mu(a) = \mu(0,0)$.

Now, since $a \neq (0,0)$, it follows that $\mathsf{val}(a) \leq 2$. So by Lemma 3.4, $\mathsf{val}(\mu((0,0))) = \mathsf{val}(\mu(a)) \leq 2$.

If $\mathsf{val}(\mu((0,0)) = 0$, it is trivial to see that G is isomorphic to K_1. Because of this, we assume that $\mathsf{val}((0,0)) \geq 1$.

Since there are no points $x \in S_n$ other than $(0,0)$ with $\mathsf{val}(x) > 2$, it follows that for all points $y \in G$, $\mathsf{val}(y) \leq 2$. That is, G is a chain.

The point $(1,1)$ has a valence of 1, and so we may conclude that it is one of the endpoints of the chain. The same applies to the points $(2,2)$ and $(3,3)$. Now, because a chain only has 2 endpoints, it follows that at least two elements of $\{(1,1),(2,2),(3,3)\}$ must map to the same point.

If $\mu((2,2)) = \mu((1,1))$, then since $(1,1)E_n(0,0)$, $\mu((2,2))D\mu((0,0))$, and so $\mu(2,1) = \mu(0,0)$. If G is not K_2, there must exist a third point x such that $x \neq \mu((2,2))$, $x \neq \mu((0,0))$ and $\mu(0,0)Dx$. This requires $\mu(2,1)Dx$, and so by the back condition, there must exist some y such that $y \neq (2,2), y \neq (0,0)$ and

$(2,1)E_n y$. No such y exists. Thus, $G = K_2$ or $\mu((2,2)) \neq \mu((1,1))$.

Thus, $\mu((2,2))$ and $\mu((1,1))$ are distinct endpoints of the chain G. From this, we can conclude that the chain G is at most 4 elements long, since that is the distance from $(2,2)$ to $(1,1)$. However, $\mu((3,3))$ is also an endpoint of the chain, and it is 5 steps away from $(1,1)$ and 6 steps away from $(2,2)$. From this, we can conclude that the length of the chain is a divisor of 6 and 4, in the case $\mu((3,3)) = \mu((1,1))$, or that the length of the chain is a divisor of both 5 and 4, in the case $\mu((3,3)) = \mu((2,2))$. Clearly, the only values that fit this are a chain of length 2, i.e K_2, or the chain of length 1, i.e. K_1. ⊣

Theorem 3.7. *For each $n \in \omega, n \geq 2$, the variety $V(\mathfrak{S}_n)$, where $\mathfrak{S}_n$ is the algebra associated with the graph S_n covers $V(\mathfrak{K}_2)$. Moreover, $\mathfrak{S}_n$ is simple and $V(\mathfrak{S}_n)$ and $V(\mathfrak{S}_m)$ are distinct when $n \neq m$. Therefore, the set of $V(\mathfrak{S}_n)_{n\in\omega}$ provide a set of countably many, finitely generated covers of $V(\mathfrak{K}_2)$ in $\Lambda(\mathsf{KTB})$.*

Proof. From Lemma 3.6, we can establish that the set of bounded morphic images of S_n is $\{K_1, K_2, S_n\}$, provided that n is at least 2. Further, S_n is a connected finite graph, and if $n \neq m$, S_n is not isomorphic to S_m.

Thus, by duality, for any $n \geq 2$ the algebra $\mathfrak{S}_n = \langle \wp(V_n); \cup, \cap, -, R_n^{-1}, \emptyset, V_n \rangle$ is a simple algebra and its only subalgebras are K_1, K_2 and itself. And if $n \neq m$, $\mathfrak{S}_n$ is not isomorphic to $\mathfrak{S}_m$

From this and Lemma 3.3, the theorem follows. ⊣

3.5 The Infinite Saw

Having seen that there are at least a countable number of covers of $V(\mathfrak{K}_2)$, we now demonstrate the existence of a continuum of such covers. This is harder, because we must move from the simple case of finite algebras and Kripke frames to the notably more complex case of infinite algebras and general frames.

We can generalise the n-spider graphs to produce the infinite spider S_ω. However, this is inadequate as the basis for an infinite algebra that covers $V(\mathfrak{K}_2)$. It is fairly easy to show that given a modal algebra $\mathfrak{A}$ based on S_ω, if A is not one of the very simple algebras equivalent to $\mathfrak{K}_1$ or $\mathfrak{K}_2$, then A is infinite, and from there, we can show that the three element chain is a member of the set $SHP_U(\mathfrak{A})$.

The problem is caused by the infinite diameter of S_ω. This means that although an algebra $\mathfrak{A}$ based on S_ω may be simple, the ultrapowers are not obliged to be "well-behaved" in any useful sense.

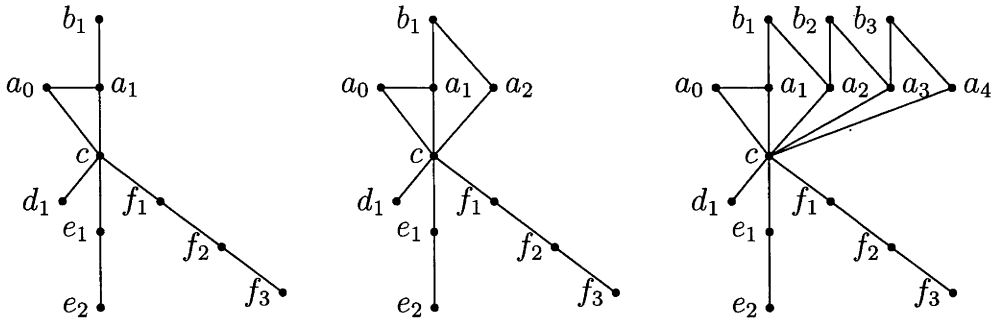

Figure 3.7: Finite graphs, with a distinct pattern

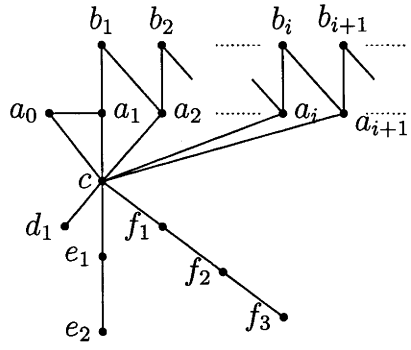

Figure 3.8: An infinite version of the finite graphs

Because of this, it makes sense to look at infinite graphs of finite diameter. After some work with automatically generating finite graphs, detailed in Appendix B, we found the three graphs of Figure 3.7. It is straightforward to verify that all 3 graphs have a diameter of 5 and the varieties of their associated algebras provide covers of $V(\mathcal{R}_2)$.

It is immediately apparent that the three graphs can be generalised to produce graphs with a longer "saw-blade" at the end, culminating in the shown in Figure 3.8.

This graph is almost what is needed. Unfortunately, is impossible to properly seperate the points e_i from the infinte section of the graph. However, it has many of the traits we want to generate an infinite cover of $V(\mathcal{R}_2)$, and it can be modified to fix this issue while preserving its "nice" traits. This produces the graph G_ω shown in Figure 3.9

Further, we can generalise the graph G_ω to produce an uncountable family of similar graphs by adding "bristles" to various points, as in Figure 3.9, which adds such a bristle to the point a_{i+1}.

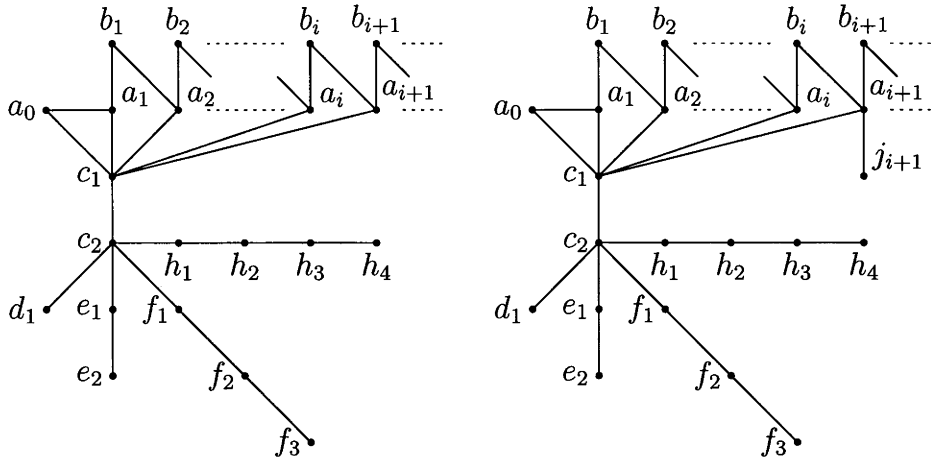

Figure 3.9: The graph G_ω, shown with and without an added point h_i

Formally, we define our family of graphs over any subset Q of positive integers that does not contain 0 or 1[2]. Given such a subset, the graph (V_Q, E_Q) is defined as follows.

We begin with sets $A = \{a_i : i \in \omega\}$, $B = \{b_i : i \in \omega \setminus \{0\}\}$, $C = \{c_1, c_2\}$, $D = \{d_1\}$, $E = \{e_1, e_2\}$, $F = \{f_1, f_2, f_3\}$, $H = \{h_1, h_2, h_3, h_4\}$ and $J_Q = \{j_i : i \in Q\}$[3]. From this, we define $V_Q = A \cup B \cup C \cup \cup D \cup E \cup F \cup H \cup J_Q$. We then define E_Q to be the symmetric, reflexive closure of the following connections:

- $c_1 E_Q c2$
- $c_2 E_Q d1, c_2 E_Q e_1, c_2 E_Q f_1, c_2 E_Q h_1$
- $e_1 E_Q e_2$
- $f_1 E_Q f_2, f_2 E_Q f_3$
- $h_1 E_Q h_2, h_2 E_Q h_3, h_3 E_Q h_4$
- $a_0 E_Q a_1$
- $\forall i \in \omega, c_1 E_Q a_i$
- $\forall i > 0, a_i E_Q b_i, b_i E_Q a_{i+1}$
- $\forall i \in Q, a_i E_Q j_i$

[2]This does not impair the uncountable nature of the possible Q, but eliminates some edge cases from the proof.

[3]To avoid confusion, we do not have sets G or I.

That is, the first 6 connections create the initial portion of the graph (the "handle" of the saw.), the next 2 create the infinite portion of the graph (the "saw-blade"), and the final condition adds bristles to various members of the graph to distinguish it from other members of the family (no saw related metaphor here). Note that S_4 from the previous section is a substructure of the initial portion of the graph.

Having defined our family of graphs, we must then show that they provide our set of infinite covers of $V(\mathfrak{K}_2)$.

Let $\mathcal{F}_Q = (V_Q, E_Q, \mathcal{I}_Q)$ be a general frame on (V_Q, E_Q), with $\mathcal{I}_Q$ the universe of the modal algebra generated by $\{h_4\}$.

Lemma 3.8. *For any subset Q of integers, where Q does not contain 0 or 1, $\mathcal{I}_Q$ consists of subsets of V_Q whose intersection with A is either finite or cofinite in A, whose intersection with B is either finite or cofinite in B and whose intersection with H_Q is either finite or cofinite in H_Q. In particular, all singletons are in $\mathcal{I}_Q$.*

Proof. Let U be the family of those subsets of V_Q whose intersection with A is either finite or cofinite in A, whose intersection with B is either finite or cofinite in B and whose intersection with J_Q is either finite or cofinite in J_Q. We must show that $U = \mathcal{I}_Q$.

It is clear that U is closed under the boolean operations and contains all singletons. Further, $\{c_1\}$ is a member of U, and so every member of U containing $\{c_1\}$ can be expressed as $\{c_1\} \cup X$ for some set X not containing c_1.

For every vertex x in V_Q other than c_1, $\mathsf{val}(x) \leq 5$. Thus, $\Diamond X \cap A$ is finite or cofinite if and only if $X \cap A$ and $X \cap B$ and $X \cap H_Q$ is finite or cofinite, and similarity for $\Diamond X \cap B$ and $\Diamond X \cap H_Q$.

If $Y = X \cup \{c_1\}$, then $\Diamond Y \cap A = A$ because $\Diamond\{c_1\} \cap A = A$, and $\Diamond Y \cap B = \Diamond X \cap B$, and $\Diamond Y \cap J_Q = \Diamond X \cap J_Q$, because $\Diamond\{c_1\} \cap (B \cup J_Q) = \emptyset$. Thus, $\Diamond Y$ is cofinite in A and finite (cofinite) in B and finite (cofinite) in H_Q iff X is finite (cofinite) in B and finite (cofinite) in A and finite (cofinite) in H_Q.

From this, we see that U is closed under the modal operator, and since $\{h_4\}$ is in U, we have $U \supseteq \mathcal{I}_Q$.

For the reverse inclusion, we need that every member of U can be generated starting from $\{h_4\}$. For this, it is sufficient to show that every singleton may be so generated.

It is trivial to see that every element of the chain $\{h_3, h_2, h_1, c_2\}$ may be generated from $\{h_4\}$. From here, $\Diamond\{c_2\} = \{c_2, c_1, e_1, d_1, f_1, h_1\}$ We can separate out c_2 and h_1, to produce the set $\{c_1, e_1, d_1, f_1\}$.

We know $\Diamond\{c_1, e_1, d_1, f_1\} \setminus \{c_1, e_1, d_1, f_1\} = A \cup \{e_2, f_2, c_2\}$. Again, we can seperate out the c_2 singleton as already having been generated, to produce $A \cup \{e_2, f_2\}$. From here, $\Diamond(A \cup \{e_2, f_2\}) \setminus (A \cup \{e_2, f_2\}) = B \cup \{e_1, f_1, f_3, c_1\}$. Complementation of this set with the set $\{c_1, e_1, d_1, f_1\}$ allows us to generate the singleton $\{d_1\}$, and so the set $\{c_1, e_1, f_1\}$, as well as the set $B \cup \{f_3\}$.

Further, $\Diamond(B \cup \{f_3\}) \setminus (B \cup \{f_3\}) = (A \setminus a_0) \cup \{f_2\}$, and since we have generated $A \cup \{e_2, f_2\}$, we can take the difference of these two sets to generate $\{a_0, e_2\}$. $\Diamond\{a_0, e_2\} = \{a_0, a_1, c_1, e_1, e_2\}$. This, along with the set $\{c_1, e_1, f_1\}$ allows us to generate the sets $\{c_1, e_1\}$ and $\{f_1\}$. From $\{f_1\}$ and $\{c_2\}$, it is trivial to generate $\{f_2\}$ and $\{f_3\}$.

Having generated $\{f_3\}$, we may seperate it from the set $B \cup \{f_3\}$, giving us the set B. $\Diamond B \setminus B = A \setminus a_0$, and $\Diamond(A \setminus a_0) \setminus (A \setminus a_0) = B \cup \{a_0, c_1\}$, and already having generated B, we have the set $\{a_0, c_1\}$. from this and the already generated set $\{c_1, e_1\}$, we can generate the individual sets $\{c_1\}, \{a_0\}, \{e_1\}$ and so $\{e_2\}$.

Now, we have successfully generated all singletons in the sets C, D, E, F and H, as well as $\{a_0\}$. From $\{a_0\}$ and $\{c_1\}$, we can trivially generate $\{a_1\}$. From here, we shall use an inductive method; We shall show if we have generated $\{a_i\}$, it is possible to generate $\{b_i\}, \{a_{i+1}\}$ and, if necessary, $\{j_i\}$.

Assume we have generated all singletons $\{a_i\}, \{b_i\}$ and $\{j_i\}$ for $i < n$, and the singleton $\{a_n\}$. Call the set of all singletons generated so far S. We consider two cases; in the first case, there exists an element $\{j_n\}$, in the second, there does not.

If there is an element $\{j_n\}$, then the set $\Diamond\{a_n\} \setminus S = \{b_n, j_n\}$. The set $\Diamond\{b_n, j_n\} \setminus (S \cup \{b_n, j_n\}) = \{a_{n+1}\}$. The set $\Diamond\{a_{n+1}\} \cap \{b_n, j_n\} = \{b_n\}$ and the set $\{b_n, j_n\} \setminus \{b_n\} = \{j_n\}$. This means that we have generated the sets $\{a_{n+1}\}, \{b_n\}$ and $\{j_n\}$.

If there is no element $\{j_n\}$, then the set $\Diamond\{a_n\} \setminus S = \{b_n\}$, and the set $\Diamond\{b_n\} \setminus \{a_n\} = \{a_{n+1}\}$.

Thus, by mathematical induction, we have shown that all singletons, and hence all members of U are in the set $\mathcal{I}_Q$, and so the two are equal. ⊣

Next, we have a proposition that should be obvious from inspection of $\mathcal{F}_Q$

Proposition 3.9. *Let $X \in \mathcal{I}_Q$ be non-empty. If $\Diamond^6 X \neq V_Q$, then either $X = \{h_4\}$ or $\neg\Diamond^6 X = \{h_4\}$*

Since $\mathcal{F}_Q$ has diameter 7, it is clear that the only X such that $\Diamond^6 X \neq V_Q$ are $\{h_4\}$ and subsets of B.

As this proposition makes clear, $\{h_4\}$ is very nearly a term definable constant. The next few lemmas make clear how easily $\{h_4\}$ can be retrieved from almost any set in $\mathcal{I}_Q$.

Let us call a set $X \in \mathcal{I}_Q$ balanced if $\Diamond X = V = \Diamond\neg X$. Clearly, any balanced set generates a 4 element subalgebra of $\mathcal{I}_Q$ isomorphic to $\mathfrak{K}_2$. Further, the set $X = V_Q$ clearly generates the two element subalgebra isomorphic to $\mathfrak{K}_1$. We will show that any other X generates $\{h_4\}$, and hence $\mathcal{I}_Q$.

Lemma 3.10. *Let $X \in \mathcal{I}_Q$ be such that $X \cap H = \emptyset$. Then X is not balanced and there are unary terms t_1, t_2, t_3, t_4 such that $t_i(X) = \{h_4\}$ for some $i \in \{1, 2, 3, 4\}$. Moreover, the following holds:*

- *If $\Diamond^6 X \neq V_Q$, then $t_4(X) = \{h_4\}$*
- *If $\Diamond^6 X = V_Q$ and $\Diamond^5 X \neq V_Q$, then $t_3(X) = \{h_4\}$*
- *If $\Diamond^5 X = V_Q$ and $\Diamond^4 X \neq V_Q$, then $t_2(X) = \{h_4\}$*
- *If $\Diamond^4 X = V_Q$ and $\Diamond^3 X \neq V_Q$, then $t_1(X) = \{h_4\}$*
- *$\Diamond^3 X \neq V_Q$.*

Proof. Let $t_1(X) = \neg\Diamond^3 X$, $t_2(X) = \neg\Diamond^4 X$, $t_3 = \neg\Diamond^5 X$ and $t_4 = \neg\Diamond^6 X$. It is obvious that if $X \cap H = \emptyset$, then $h_2 \notin \Diamond X$, and so X is not balanced, and $h_4 \notin \Diamond^3 X$.

By inspection of the frame, it is immediately seen that $\neg\Diamond^3\{c_2\} = \{h_4\}$, and for any X such that $X \cap H = \emptyset$, there exists an $n < 4$ such that $\Diamond^n X$ contains c_2, but no element of H. Then for this n, $\neg\Diamond^{3+n} X$ is one of our terms t_i, and $t_i(X) = \{h_4\}$.

From here, the lemma follows. ⊣

Next, we must consider the case of generating $\{h_4\}$ from sets $X \in \mathcal{I}_Q$ with $X \cap H \neq \emptyset$. We do this by showing that unless X is balanced, or $X = V_Q$, we can generate a set Y such that $Y \cap H = \emptyset$, and $Y \neq \emptyset$. With such a Y, we can apply Lemma 3.10.

Lemma 3.11. *Let X be unbalanced, with $X \cap H \neq \emptyset$ and $X \neq V_Q$. Then we can generate a non-empty set Y such that $Y \cap H = \emptyset$ using one of a finite set of terms.*

Proof. To generate such a set Y, it is sufficient to generate a set Z such that $Z \supseteq H$ and $Z \neq V_Q$. Then our Y is simply $\neg Z$. Without loss of generality, assume that $\{h_1\}$ is in X (Otherwise, we could apply these arguments to $\neg X$ instead). From here, we have an argument by cases, for the various values of $X \cap H$:

- $X \cap H = \{h_1\}$: Suppose $c_2 \in X$. Then $\Diamond^2 X \supset A \cup C \cup D \cup E \cup \{f_1, f_2\}$, however, $h_4 \notin \Diamond^2 X$. Thus $\neg\Diamond^2 X \subseteq B \cup \{f_3, h_4\}$. Clearly, $\Diamond^3\neg\Diamond^2 X \supseteq H$ and $\Diamond^3\neg\Diamond^2 X \cap E = \emptyset$. Thus, $\neg\Diamond^3\neg\Diamond^2 X$ is a non-empty set and $\neg\Diamond^3\neg\Diamond^2 X \cap H = \emptyset$.

  Conversely, suppose $c_2 \notin X$. Then we have two cases. In the first case, $\Diamond^3 X \neq V_Q$. In this case, $\neg\Diamond^3 X$ is the desired Y. Conversely, $\Diamond^3 X = V_Q$ implies $\Diamond^3 X \supset F$. Combined with $c_2 \notin X$, this gives us that there is some i such that $f_i \in X$, which in turn give us that $\Diamond^2 X \supset F$. From this, we may conclude that $\{f_3\} \notin \Diamond^3\neg\Diamond^2 X$. Thus, $\neg\Diamond^3\neg\Diamond^2 X$ is a non-empty set and $\neg\Diamond^3\neg\Diamond^2 X \cap H = \emptyset$.

- $X \cap H = \{h_1, h_2\}$: If $c_2 \in X$, then $\neg\Diamond\neg X \cap H = \{h_1\}$, which case has already been covered.

  Suppose $c_2 \notin X$. Now, $\Diamond^2 X \supseteq H$, so either $\Diamond^2 X = V_Q$ or $\neg\Diamond^2 X$ is our desired Y. If $\Diamond^2 X = V_Q$, this means $e_2 \in \Diamond^2 X$, which combined with $c_2 \notin X$ means that some e_i is in X, and so $\Diamond X \supset E$. From this, $\neg\Diamond X$ contains h_4, but not e_1, e_2 or c_2.

  Now, if $c_2 \in \Diamond\neg\Diamond X$, then $\Diamond^2\neg\Diamond X \supset H$, but $e_2 \notin \Diamond^2\neg\Diamond X$, so our desired Y is $\neg\Diamond^2\neg\Diamond X$. On the other hand, if $c_2 \notin \Diamond\neg\Diamond X$, then $e_2 \notin \Diamond^3\neg\Diamond X$ and $\Diamond^3\neg\Diamond X \supseteq H$, so our desired Y is $\neg\Diamond^3\neg\Diamond X$.

- $X \cap H = \{h_1, h_3\}$: Here, $\Diamond X \cap H = H = \Diamond\neg X$. If $\Diamond X = V_Q = \Diamond\neg X$, then X is a balanced set, contrary to our assumptions. Thus, our desired Y is either $\neg\Diamond X$ or $\neg\Diamond\neg X$.

- $X \cap H = \{h_1, h_4\}$:Here, $\Diamond X \cap H = H = \Diamond\neg X$. If $\Diamond X = V_Q = \Diamond\neg X$, then X is a balanced set, contrary to our assumptions. Thus, our desired Y is either $\neg\Diamond X$ or $\neg\Diamond\neg X$.

- $X \cap H = \{h_1, h_2, h_3\}$: If $c_2 \in X$, then $\neg\Diamond\neg X \cap H = \{h_1, h_2\}$, a case already covered.

Otherwise, $\neg\Diamond\neg X \cap H = \{h_2\}$, and for all points y such that $c_2 \in \Diamond\{y\}$, $y \notin \neg\Diamond\neg X$, and as such, neither d_1 or c_2 are in $\Diamond\neg\Diamond\neg X$. Thus, we have $\Diamond^2\neg\Diamond\neg X$ contains all h_i, but does not contain d_1. As such, $\neg\Diamond^2\neg\Diamond\neg X$ is our desired Y.

- $X \cap H = \{h_1, h_2, h_4\}$: If $c_2 \in X$, then $\neg\Diamond\neg X = \{h_1, h_2\}$, a case already covered.

  If $c_2 \in \neg X$, then $\Diamond\neg X \cap H = H = \Diamond X \cap H$. If $\Diamond X = V_Q = \Diamond\neg X$, then X is a balanced set, contrary to our assumptions. Thus, our desired Y is either $\neg\Diamond X$ or $\neg\Diamond\neg X$.

- $X \cap H = \{h_1, h_3, h_4\}$: Here, $\Diamond\neg X \cap H = \{h_1, h_2, h_3\}$, a case already covered.

- $X \cap H = \{h_1, h_2, h_3, h_4\}$: Since it is a condition of the lemma that $X \neq V_Q$, our desired Y is simply $\neg X$

⊣

Given Lemma 3.10 and Lemma 3.11, we may combine them for the following deduction:

Lemma 3.12. *There are finitely many terms $t_1 \ldots t_k$ such that for any unbalanced set X, there is a $t_i, 1 \leq i \leq k$, such that $t_i(X) = \{h_4\}$.*

Now, we have a proposition that follows directly from inspection of the frame:

Proposition 3.13. *$\{h_4\}$ is the unique atom a in $\mathcal{I}_Q$ such that all of $\Diamond a \setminus a$, $\Diamond^2 a \setminus \Diamond a$, $\Diamond^3 a \setminus \Diamond^2 a$ and $\Diamond^4 a \setminus \Diamond^3 a$ are all atoms.*

Since it is possible to express the property of being an atom with a first order algebraic formula, it is also possible to express the property of being the atom $\{h_h\}$ with a first order algebraic formula. Let us call this formula H_4; that is:

Definition 3.14. For any $X \subseteq V_Q$, we have $H_4(X)$ if and only if $X = \{h_4\}$.

Let $\mathfrak{A}_Q$ be the modal algebra dual to the frame $\mathcal{F}_Q$.

Now, since the frame has finite diameter, $V(\mathfrak{A}_Q)$ is a discriminator variety, which gives us a number of nice properties, including that subdirectly irreducible algebras are simple [8].

Lemma 3.15. *Let $\mathfrak{B} \in V(\mathfrak{A}_Q)$ be a simple algebra. If $\mathfrak{B}$ is not isomorphic to $\mathfrak{K}_1$ or $\mathfrak{K}_2$, then $\mathfrak{A}_\Omega$ is a subalgebra of $\mathfrak{B}$.*

Proof. By Jónsson's Lemma [25] and $V(\mathfrak{A}_Q)$ being a discriminator variety, $\mathfrak{B} \in \mathbf{SP_U}(\mathfrak{A}_Q)$. Since $\mathfrak{B}$ has more than 4 elements, there is an element $b \in \mathfrak{B}$ with $\Diamond b \neq 1$. Then, by Lemma 3.12, there is some term t_i such that $B \models H_4(t_i(b))$, and by definition, the element $t_i(b)$ will generate an isomorphic copy of $\mathfrak{A}_Q$. Thus, $\mathfrak{A}_Q$ is a subalgebra of $\mathfrak{B}$. ⊣

Now, one final lemma, and we may produce our theorem.

Lemma 3.16. *For distinct Q, Q', the dual algebras of $\mathcal{F}_Q$ and $\mathcal{F}_{Q'}$ are non-isomorphic.*

Proof. Since the algebras involved possess a discriminator term, atoms are first order-definable. As such, any isomorphism between the two must map atoms to atoms.

Without loss of generality, assume that there exists some $i \in Q \setminus Q'$. Then $\Diamond\{a_i\}$ in $\mathcal{F}_Q$ is the join of five atoms $(\{a_i, b_i, b_{i-1}, c_1, h_i\})$, while $\Diamond\{a_i\}$ in $\mathcal{F}_{Q'}$ is the join of four atoms $(\{a_i, b_i, b_{i-1}, c_1\})$.

But an isomorphism between the dual algebras must provide an isomorphism from one set to the other, so there is a contradiction. ⊣

From Lemmas 3.15 and 3.16, we have:

Theorem 3.17. *For any subset Q of the natural numbers, not containing 0 or 1, the variety of the algebra dual to the frame $\mathcal{F}_Q$ covers $V(\mathfrak{K}_2)$ in the lattice $\Lambda(\mathsf{KTB})$.*

For any distinct subsets Q, R of the natural numbers, the covering varieties created thus are non-isomorphic.

As such, $V(\mathfrak{K}_2)$ has an uncountable number of covers in the lattice $\Lambda(\mathsf{KTB})$.

Corollary 3.18. *Each of the general frames $\mathcal{F}_Q$ validates a symmetric, reflexive logic with only two consistent normal extensions.*

Corollary 3.19. *There are an uncountable number of logics L in $NExt(\mathbf{KTB})$ such that L has only two consistent normal extensions. These extensions are the logic of the two element chain and the logic of the single reflexive point.*

Chapter 4

Normal Forms

4.1 Introduction

In the study of modal logics, one recurring question is the Kripke completeness of logics, as outlined in Chapter 2. When given a logic $\mathbf{L}$ and a formula $\varphi \notin \mathbf{L}$, the key to Kripke completeness is being able to construct an appropriate frame $\mathcal{F}$, such that $\mathcal{F} \models \mathbf{L}$, but $\mathcal{F} \not\models \varphi$.

There are two methods commonly used to show Kripke completeness. The first of these is semantic tableaux, which provide a constructive proof of completeness. However, Fine [16] claims that tableau methods are "not elegant".

On the other hand, maximally consistent theories provide an elegant way to show Kripke completeness. However, they are not constructive, which is something of a drawback.

As an attempt to create a method that is both elegant and constructive, Fine [16] presents normal forms. However, while normal forms can be elegant, it is also the case that there are certain common pitfalls that occur when trying to use normal forms. This chapter reviews the definitions and methods needed to produce results using normal forms, and provides several example results, pointing out some pitfalls that can occur. While tableau methods can be automated, using normal forms for an automated decision procedure is not really practical. The complexity of a normal form explodes rapidly compared to the complexity of the original formula.

As pointed out in Chapter 2, the finite model property (FMP) is a stronger property than Kripke completeness. As such, if you can show a logic has the finite model property, Kripke completeness follows. Since normal forms are extremely well suited to the generation of finite models, we frequently use it to show the

FMP. In this case, we can assume as a corollary that the logic is Kripke complete.

Note that this work was done without knowledge of the paper [34] which also covers the method of normal forms, and reworks the definitions in light of other work, such as [27]. The paper [18] also discusses normal forms, and takes a more algebraic approach.

4.2 Preliminary Definitions

An important abuse of notation when discussing normal forms is the use of formulas to name points in a Kripke frame. That is, when we speak of a point A in a model, we associate with A some formula α. For the most part, it is desirable that the formula α is true at the point A. We write this $A \vDash \alpha$. We shall be consistent in associating formulas α, β with points A, B. Indeed, the great strength of normal forms is this geometric approach, where each normal form is in some sense simultaneously a formula of the logic and a point in the frame.

When working with normal forms, we assume a fixed finite set $Q = \{q_0 \dots q_h\}$ of propositional variables, and suppose that all formulas are constructed from the variables of Q. The exact size of Q is irrelevant. When dealing with a particular formula, we assume that Q is sufficiently large to contain all variables needed in that formula. Given this set Q, it is clear that propositional variables are ordered, and it is trivial to extend this to a standard order on formulas.

We define the degree, $deg(\varphi)$, of a modal formula as follows:

Definition 4.1 (Degree).

$$deg(q_i) = deg(\bot) = 0$$

$$deg(\neg\psi) = deg(\psi)$$

$$deg(\psi \vee \chi) = \max(deg(\psi), deg(\chi))$$

$$deg(\psi \wedge \chi) = \max(deg(\psi), deg(\chi))$$

$$deg(\Diamond\psi) = deg(\Box\psi) = 1 + deg(\psi)$$

.

This definition allows us to perform induction over the degree of a modal formula, something we will frequently exploit when using normal forms.

The set of normal forms itself is defined using recursion over the degree of the normal form.

Definition 4.2 (Normal Forms). F_0, the set of **normal forms of degree zero** is the set of formulas of the form $\bigwedge_{i=0}^{h} \pi_i q_i$, where each π_i is either blank or $\neg$, and the q_i range over all the members of our variable set Q.

For $n > 0$, we define F_n, the set of **normal forms of degree** n as the set of formulas of the form $\beta \wedge \bigwedge_{i=0}^{m} \pi_i \Diamond \alpha_i$, where $\beta \in F_0$, and π_i is either blank or $\neg$, and the α_i are all m elements of F_{n-1} in a standard order.

Since a normal form of degree n contains $\Diamond\alpha$ or $\neg\Diamond\alpha$ for every normal form α of degree $n-1$, when referring to a normal form, it is sufficient to list only those normal forms that occur in the form $\Diamond\alpha$, and assume that if an element of F_{n-1} is not listed, then it occurs in the form $\neg\Diamond\alpha$. We will write this as $\alpha = \beta \wedge \Diamond\{\gamma_i\}$ and refer to this as the $\Diamond$set notation.

When we must discuss normal forms of multiple degrees, we shall use a consistent formula symbol for elements of each F_n, and use subscripts to distinguish elements. That is, if we need to have formulas from F_n, F_{n-1} and F_{n-2} for some n, then elements of F_n shall be $\alpha_1, \alpha_2 \ldots$, elements of F_{n-1} shall be $\beta_1, \beta_2 \ldots$ and elements of F_{n-1} shall be $\gamma_1, \gamma_2 \ldots$.

Now, while an individual normal form is overly specific for most purposes, it is proven in [16] that any formula φ is equivalent in **K** to either $\bot$ or a disjunction of normal forms of the same degree as φ. Furthermore, it is a natural corollary of the proof method that a given disjunction of normal forms is unique.

A normal form may be seen as describing a section of a model out to a given depth; It states which variables are true at a point, and which variables are true at related points, and so forth, until reaching the limit of the normal form's degree.

An important component in a normal form α is the degree zero term that describes the current point. We call this the leading term and denote it α^l.

Intuitively, consider the normal form $\alpha = \alpha^l \wedge \Diamond\{\beta_1, \beta_2, \beta_3\}$, where $\beta_1 = \beta_1^l \wedge \Diamond\{\gamma_1, \gamma_2\}, \beta_2 = \beta_2^l \wedge \Diamond\{\gamma_2, \gamma_3\}, \beta_3 = \beta_3^l \wedge \Diamond\{\gamma_2, \gamma_3, \gamma_4\}$ and $\alpha^l, \beta_1^l, \beta_2^l, \beta_3^l, \gamma_1, \ldots \gamma_4$ are all distinct normal forms of degree zero. From this form α, we can create the frame $\mathcal{F}_\alpha$ in Figure 4.1. With each formula $\alpha, \beta_i, \gamma_i$, we associate a point A, B_i, C_i, and if for any pair of formulas φ, ψ, we have $\varphi \rightarrow \Diamond\psi$, we add a directed edge from the associated point of φ to the associated point of ψ. If we desire to turn this frame into a model such that $A \vDash \alpha$, then we can do so by allowing the leading terms to define the variables true at each point. For formalisation of this process, see Section 4.3.

One basic result on normal forms is that normal forms of the same degree disagree. That is, if $deg(\alpha) = deg(\beta)$, and $\alpha \neq \beta$, then $\mathbf{K} \vDash \alpha \rightarrow \neg\beta$. When we

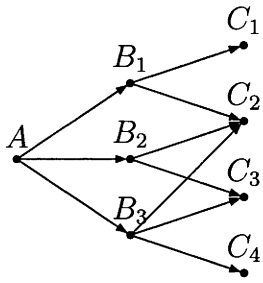

Figure 4.1: Frame $\mathcal{F}_\alpha$

discuss normal forms, there are many results of the form $\mathbf{K} \vDash \alpha \to \beta$. For the most part, we shall write only $\alpha \to \beta$, and leave the choice of logic implicit.

On normal forms, we define some binary relations:

- $\alpha > \beta$ iff $\Diamond\beta$ is a conjunct of α
- $\alpha >^n \beta$ iff $\exists\gamma_1 \ldots \gamma_{n-1}$, such that $\alpha > \gamma_1 > \ldots > \gamma_{n-1} > \beta$
- $\alpha \gg \beta$ iff $\exists n$ such that $\alpha >^n \beta$
- $\alpha \succ \beta$ iff $\forall\gamma$, $\beta > \gamma \Rightarrow \alpha > \gamma$

If we continue to use the intuition of a normal form describing a part of the model, then $\alpha >^n \beta$ tells us that our A point sees a B point in n steps, and $\alpha \gg \beta$ tells us that a B point is within the n-th upward closure of A for some n.

The relation $\alpha \succ \beta$ is less intuitive. However, it can be considered to mean that A sees every point that B sees. In a transitive frame, if our A point is related to the B point, then $\succ$ follows by transitivity.

Lastly, we define a way to move between different degrees of normal forms.

Firstly, we take the correlate, α' which can be considered a zooming in operation; it is the unique normal form of degree one lower than α such that $\mathbf{K} \vDash \alpha \to \alpha'$.

Definition 4.3 (Correlate). The **correlate** α' of a normal form α is defined recursively.

$$\alpha' = \begin{cases} \alpha^l & \text{if } deg(\alpha) = 1 \\ \alpha^l \wedge \bigwedge_{\beta \in \Delta_\alpha} \Diamond\beta \wedge \bigwedge_{\beta \in F_{n-1} \setminus \Delta_\alpha} \neg\Diamond\beta & \text{otherwise.} \end{cases}$$

where $\Delta_\alpha = \{\beta' : \alpha > \beta\}$

More simply, using our previous "$\Diamond$set" notation, if $\alpha = \alpha^l \wedge \Diamond\{\beta_i\}$, then $\alpha' = \alpha^l \wedge \Diamond\{\beta_i'\}$

For normal forms of degree 0, various relations and the correlate are undefined. For most purposes this is unfortunate, but largely irrelevant. In the search for elegance, it is possible to introduce $\top$ as a normal form of 'minimal degree', and claim that if $deg(\alpha) = 0$, $\alpha' = \top$, and $\alpha > \top$. Most proofs will continue to work without modification, and some are even simplified. This does however, lead to assuming that within the logic under consideration, $\alpha \rightarrow \Diamond\top$ holds for all normal forms α. This is fine when working with seriality, or stronger axioms such as reflexivity. However, if it is necessary that there be a normal form α such that $\{\beta : \alpha > \beta\} = \emptyset$, then this definition can cause problems. Thus, we do not use this definition. However, it can be useful in such cases as deontic logic (see Fine [16]).

While knowing α is enough to determine α', even brief examination of the case where $deg(\alpha) = 1$ is enough to demonstrate that there can be many normal forms β with $deg(\beta) = 2$ and $\beta' = \alpha$. Thus, because of this discarding of information, there can be no convenient inverse operation $\circ$ to the correlate, such that $(\alpha')^\circ = \alpha$. However, we can define operations $\circ$ such that $(\alpha^\circ)' = \alpha$. Since such counter-correlate operations can be useful, we define two such operations here:

Definition 4.4 (Maximal Counter-correlate)**.** We define the **maximal counter-correlate** α° of α as the greatest (under $\succ$) normal form β such that $\beta' = \alpha$. Equivalently, if $deg(\alpha) = n$:

$$\alpha^\circ = \alpha^l \wedge \bigwedge_{\beta \in \Sigma_\alpha} \Diamond\beta \wedge \bigwedge_{\beta \in F_n \setminus \Sigma_\alpha} \neg\Diamond\beta$$

where $\Sigma_\alpha = \{\beta : \alpha > \beta'\}$.

Definition 4.5 (Limited Counter-correlate)**.** We define the **limited counter-correlate** of α to be the normal form β that "sees" nothing that α does not see:

$$\alpha^- = \begin{cases} \alpha^l \wedge \Diamond\emptyset & \text{if } deg(\alpha) = 0 \\ \alpha^l \wedge \Diamond\{\beta^- : \alpha > \beta\} & \text{otherwise.} \end{cases}$$

The important property of counter-correlates is that $(\alpha^\circ)' = (\alpha^-)' = \alpha$. Thus, since we already know $\alpha \rightarrow \alpha'$, we have:

Proposition 4.6. *$\alpha^\circ \rightarrow \alpha$, and $\alpha^- \rightarrow \alpha$.*

4.3 Standard model constructions

It has been stated a couple of times already that a normal form can be considered as describing a point in a model. In this section, we outline three standard ways to go from normal forms to models.

In our model, each world is associated with a specific normal form. We name worlds for their associated normal form. From these associations, we define the standard valuation on a frame:

Definition 4.7 (Standard Valuation)**.** The **standard valuation** V_S on a frame $\mathcal{F}$, whose worlds are associated with normal forms, is the valuation such that for any world A, associated with α, we have $\langle \mathcal{F}, V_S \rangle, A \vDash \alpha^l$.

To interpret this, we recall that α^l, a normal form of degree 0, has the form $\bigwedge_{i=0}^{h} \pi_i q_i$, where π_i is either blank or $\neg$. If π_i is blank, then A is in $V_S(q_i)$. If π_i is $\neg$, then A is not in $V_S(q_i)$. From the definition, we have:

Proposition 4.8. *$\alpha \to q_i$ iff $A \in V_S(q_i)$.*

4.3.1 The graded model

The graded model $\mathfrak{A}_\alpha$ is generated to make a specific normal form α true. We take A, associated with α as its root, and add enough extra points, of increasingly lower degree, to make $A \vDash \alpha$ true. Formally:

Let our set of normal forms be $\{\alpha\} \cup \{\beta : \alpha \gg \beta\}$, and associate with each normal form a point. The point A shall be associated with α, and for each β_i, we associate a point B_i. Then let $W_\alpha = \{A, B_1, \ldots, B_i\}$. That is, our frame will contain the worlds associated with the normal forms contained in α. We define the relation R_α by saying there is a connection $BR_\alpha C$ if, with the normal forms β, γ associated with B and C, we have $\beta > \gamma$. That is, let $R_\alpha = \{(B, C) \in W_\alpha^2 : \beta > \gamma\}$.

From this, let $\mathcal{F}_\alpha = \langle W_\alpha, R_\alpha \rangle$, and $\mathfrak{A}_A = \langle \mathcal{F}_A, V_S \rangle$, with V_S being the standard valuation outlined earlier.

We call this $\mathfrak{A}_\alpha$ the graded model for α, as the degree of the normal forms at each point gets lower as we progress down the model.

For an example of constructing a graded model, consider the earlier example of Figure 4.1. The construction used to produce that frame is the same construction used to produce graded models. Having defined the construction of the graded frame for a formula α, we may also see that the limited counter-correlate will

define the same graded frame. Figure 4.2 shows the frames $\mathcal{F}_\alpha$ and $\mathcal{F}_{\alpha^-}$, where α is defined as for Figure 4.1.

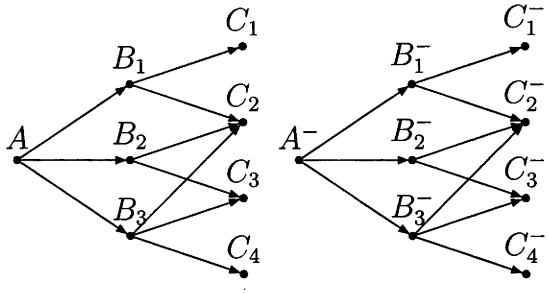

Figure 4.2: Graded frame $\mathcal{F}_\alpha$ and the counter-correlate frame $\mathcal{F}_{\alpha^-}$.

In Fine [16] and Moss [34], the graded model is defined starting with the set of all normal forms of degree n. Essentially, our definition is the submodel generated by A.

4.3.2 The graded tree

The graded tree is a variant of the graded submodel, created to produce a tree with many of the same basic properties. A normal form α could contain a trio of normal forms β_1, β_2, γ, with the property $\beta_1 > \gamma$ and $\beta_2 > \gamma$. In the graded model $\mathfrak{A}_\alpha$, this trio of normal forms would be represented by three worlds, with $B_1 R_\alpha C$ and $B_2 R_\alpha C$.

On the other hand, in a graded tree, we would split the world C into two worlds C_{β_1} and C_{β_2}, and have $B_1 R C_{\beta_1}$ and $B_2 R C_{\beta_2}$. For the associated model, we would keep an identical valuation for both C_{β_1} and C_{β_2}.

We construct the graded tree recursively. For a normal form α of degree 0, the graded tree $\mathfrak{T}_\alpha$ is a single point, identical to the graded frame.

For a normal form α of degree n, construct the trees $\mathfrak{T}_\beta$ for all β such that $\alpha > \beta$. Then, take the disjoint union of these trees, distinguishing otherwise identical world names with subscripts, and add a world A, with ARB for the root node B of each tree $\mathfrak{T}_\beta$.

The graded tree in Figure 4.3 is based on the same normal form A as the frame in Figure 4.1.

The graded tree is very similar to the graded model, and can be used for many of the same proofs. However, it has the property that for a point X, there is only a single point Y such that YRX. For an example of how this is used, see the proof that $\mathbf{KTB} \bigoplus alt_n$ has the finite model property in Section 4.5.1.

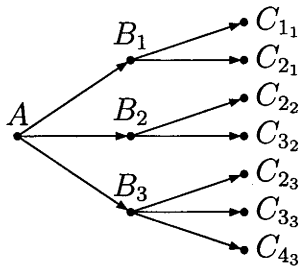

Figure 4.3: The graded tree $\mathfrak{T}_A$

4.3.3 The ungraded model

The graded model works by taking a normal form α and then the set of normal forms $\{\beta : \alpha > \beta\}$, and the set of normal forms $\{\gamma : \alpha >^2 \gamma\}$, and so gradually working down to points of degree zero. For some logics, this is inadequate. The graded model is always irreflexive and asymmetric. While modifications can get around this (trivially so, in the case of irreflexivity) it is nonetheless desirable to have a model where these properties fall out naturally.

For an example of the application of such a model, see [16], where the ungraded model is used to show the FMP for transitivity - the simple graded model is inadequate to distinguish **K4**, extending **K** by adding the axiom $\Box p \rightarrow \Box\Box p$ and **GL**, extending **K4** by adding the axiom $\Box(\Box p \rightarrow p) \rightarrow \Box p$.

To define $\mathfrak{A}_n$, the ungraded model of degree n, we take a world set containing a world for every normal form of degree n, and define the relation set by saying a world has a connection to some other world if, for the associated normal forms α_1, α_2, we have $\alpha_1 \rightarrow \Diamond\alpha_2'$. That is, α_1 sees the correlate of α_2.

As before, we construct the model $\mathfrak{A}_n$ from $\mathcal{F}_n = \langle W_n, R_n \rangle$ and the standard valuation: $\mathfrak{A}_n = \langle \mathcal{F}_n, V_S \rangle$.

Assuming that F_n, the set of normal forms of degree n, contains m formulas and is indexed as $\alpha_i, 0 < i \leq m$, then we define the world set $W_n = \{A_i : 0 < i \leq m\}$. For each i, we associate the world A_i and the formula α_i. Then for the relation set we have $R_n = \{(A_i, A_j) \in W_n^2 : \alpha_i > \alpha_j'\}$.

While for simple normal forms, the graded model may be easy to draw, the ungraded model includes *all* the normal forms of a given degree. This number grows extremely rapidly, and quickly becomes unfeasibly large to capture, especially when the set of possible variables Q is large.

4.4 Some basic results

The following results are all stated without proof. Most are trivial, and proofs for the others can be found in Fine [16] or Miyazaki [32]:

- $\alpha > \beta \Rightarrow \alpha' > \beta'$
- $\alpha \succ \beta \Rightarrow \alpha' \succ \beta'$
- $deg(\alpha) = deg(\beta)\ \&\ \alpha \neq \beta \Rightarrow \alpha \rightarrow \neg\beta$
- $\bigvee F_n = \top$
- $\mathbf{K} \vDash \alpha \rightarrow \alpha'$
- $(\alpha^\circ)' = \alpha$
- $\mathbf{K} \vDash \alpha^\circ \rightarrow \alpha$
- $\alpha = \beta' \Rightarrow \alpha^\circ \succ \beta$
- $\alpha > \beta \Rightarrow \alpha^\circ > \beta^\circ$
- $\langle \mathfrak{A}_\alpha, B \rangle \vDash \beta$
- $\langle \mathfrak{A}_n, A_i \rangle \vDash \alpha_i$.

4.5 Application

One of the basic applications of normal forms is to produce constructive proofs of Kripke completeness, and one of the easiest ways to do this is to show the existence of the finite model property (FMP). That is, we show how to construct finite Kripke models that refute formulas not within the logic. Key to this procedure is the theorem of Fine [16], which states that all modal formulas φ, with $deg(\varphi) = n$ are equivalent in $\mathbf{K}$ to either $\bot$ or to a disjunction of normal forms of degree n.

The first stage in using normal forms to demonstrate the FMP for a logic $\mathbf{K} \bigoplus \psi$ is to define the set of ψ-suitable normal forms. In the logic $\mathbf{K} \bigoplus \psi$, normal forms that are not ψ-suitable are equivalent to $\bot$. Given this equivalence, and the prior theorem, it is apparent that any formula φ of degree n is equivalent in $\mathbf{K} \bigoplus \psi$ to some disjunction of ψ-suitable normal forms of degree n. The equivalence to a disjunction of normal forms is a theorem of $\mathbf{K}$, and the normal forms that are not ψ-suitable play no role in the disjunction's truth in $\mathbf{K} \bigoplus \psi$.

Thus, if we can successfully define ψ-suitable normal forms, we can state that all formulas φ, of degree n are equivalent in $\mathbf{K} \bigoplus \psi$ to $\bot$ or a disjunction of ψ-suitable normal forms of degree n.

The second stage in demonstrating that a logic has the FMP is to demonstrate a way to take a ψ-suitable normal form α and produce a model $\langle \mathcal{F}_\alpha, V_S \rangle$, where $\mathcal{F}_\alpha \vDash \psi$, with some world A in this model such that $(\langle \mathcal{F}_\alpha, V_S \rangle, A) \vDash \alpha$.

Given this, from the previous observation, if φ is not in $\mathbf{K} \bigoplus \psi$, then $\neg\varphi$ is not equivalent to $\bot$ in $\mathbf{K} \bigoplus \psi$. So, it must be equivalent to some disjunction of ψ-suitable normal forms. Thus, we can take one ψ-suitable normal form, α, from the disjunction, and produce a ψ-model that makes α true at a point, and since $\alpha \rightarrow \neg\varphi$, we have $\mathfrak{A}_\alpha \nvDash \varphi$.

Generally, to derive the relevant model, it is sufficient to use some variant of the graded or ungraded model. When using the ungraded model, it is essential to restrict W_n to only contain ψ-suitable normal forms, and can sometimes be necessary to restrict R_n as well.[1]

4.5.1 Examples

In [16], normal forms are used to show the FMP for **K**, **T**, **K4** and all uniform extensions of **D**. Here, we use the technique outlined above to show the FMP for some other common logics.

First, we reproduce the result for **T**, as several examples of the finite model property generalise easily from **L** to $\mathbf{L} \bigoplus T$, and understanding how **T** works is important for doing this.

Modal logic T

$\mathbf{T} = \mathbf{K} \bigoplus T$ where $T = \Box p \rightarrow p$. We define **T**-suitable normal forms:

Definition 4.9 (**T**-suitable normal forms)**.** A normal form α is **T-suitable** if

- $deg(\alpha) = 0$ or
- $\alpha > \alpha'$ and
- $\forall \beta, \alpha > \beta$ implies β is **T**-suitable.

[1] For example, in [16], FMP for K4 is shown using a restriction of R_n to $\{\langle A_i, A_j \rangle \in W_n^2 : \alpha_i > \alpha'_j \ \& \ \alpha_i \succ \alpha_j\}$.

Since $\alpha \rightarrow \alpha'$, it is trivial to see that for any normal form α, either $\alpha \rightarrow \Diamond\alpha'$ (and so $\Diamond\alpha'$ is a conjunct of α) or $\mathbf{T} \vDash \alpha \leftrightarrow \bot$.

The other two conditions of **T**-suitability, that all degree 0 normal forms are **T**-suitable, and a normal form of degree $n \geq 0$ is only **T**-suitable if all β such that $\alpha > \beta$ are **T**-suitable, are general conditions. Similar conditions apply to **L**-suitability for all logics **L**.

Since degree 0 normal forms do not contain any modal operators, then axioms that make statements about modal operators are irrelevant to their potential truth. And if $\alpha > \beta$, and β is not **L**-suitable, then $\mathbf{L} \vDash \beta \leftrightarrow \bot$, and so α contains a conjunct equivalent to $\Diamond\bot$, which makes α equivalent to $\bot$.

For an appropriate model for **T**-suitable logics, Fine [16] uses the ungraded model, with no modifications beyond restricting the frame $\langle W_n, R_n \rangle$ for a normal form of degree n, to **T**-suitable normal forms. It is simple to show that this still has the property $\langle \mathfrak{A}_n, A_i \rangle \vDash \alpha_i$, and since $\alpha_i > \alpha_i'$ for all **T** suitable α, we also have $(A_i, A_i) \in R_n$.

It is also possible to modify the graded models to produce a satisfying model for a **T**-suitable α; it is trivial to show that if α is **T**-suitable, then the reflexive closure of $\mathfrak{A}_\alpha$ maintains the property that $\langle \mathfrak{A}_\alpha, A \rangle \vDash \alpha$.

Directed Modal logics

A directed frame has the following property:

$$\forall x, y, z(xRy \wedge xRz \wedge y \neq z \rightarrow \exists u(yRu \wedge zRu))$$

For frames, it can be shown that $\mathcal{F} \vDash \Diamond(\Box p \wedge q) \rightarrow \Box(\Diamond p \vee q)$ iff $\mathcal{F}$ is directed. We call the formula $\Diamond(\Box p \wedge q) \rightarrow \Box(\Diamond p \vee q)$ *dir*, and show that $\mathbf{K} \bigoplus dir$ has the finite model property.

We define directed normal forms:

Definition 4.10. A normal form α is **directed** if

- $deg(\alpha) = 0$ or $deg(\alpha) = 1$
- $\alpha > \beta_1$ & $\alpha > \beta_2$ & $\beta_1 \neq \beta_2 \Rightarrow \exists\gamma$ s.t. $\beta_1 > \gamma$ & $\beta_2 > \gamma$
- $\forall\beta, \alpha > \beta \Rightarrow \beta$ is directed.

Here, the graded frame $\mathfrak{A}_\alpha$ works almost without modification. However, to preserve the directedness of the frame at the lowest points, it is necessary to add

a world a to W_A, and modify R_α to include $\langle A, a\rangle$ for all A whose associated formula α has degree 0[2]. Verifying that this frame is directed is trivial.

To show that non-directed normal forms are equivalent to $\bot$ in $\mathbf{K} \bigoplus dir$, we may reason like this:

Suppose that α is not directed. This means $\exists \beta_1, \beta_2$ such that $\alpha > \beta_1, \alpha > \beta_2, \beta_1 \neq \beta_2$ and $\beta_1 > \gamma$ implies $\beta_2 \not> \gamma$.

Take $\varphi = \neg \bigvee \{\gamma : \beta > \gamma\}$. Now $\beta_2 \rightarrow \Box\varphi$ since otherwise, $\beta_2 \rightarrow \Diamond\gamma$, for some $\beta_1 > \gamma$, a contradiction.

Thus, $\alpha \rightarrow \Diamond(\Box\varphi \wedge \beta_2)$. Therefore, by the axiom *dir*, under substitution, we have $\alpha \rightarrow \Box(\Diamond\varphi \vee \beta_2)$. Now, $\beta_1 \neq \beta_2$, so $\beta_1 \rightarrow \neg\beta_2$, and by the definition of φ, $\beta_1 \rightarrow \Diamond\gamma$ means $\gamma \rightarrow \neg\varphi$. So $B \rightarrow \neg\Diamond\varphi$.

Therefore, we have $\alpha \rightarrow \Box(\Diamond\varphi \vee \beta_2)$ and $\alpha \rightarrow \Diamond(\neg\beta_2 \wedge \neg\Diamond\varphi)$, a contradiction. We may conclude $\mathbf{K} \bigoplus dir \vDash A \leftrightarrow \bot$.

Connected modal logics

A connected frame $\mathcal{F}$ has the following property:

$$\forall x, y, z(xRy \wedge xRz \wedge y \neq z \rightarrow yRz \vee zRy)$$

Modally, this is equivalent to the statement $\mathcal{F} \vDash \Box(p \wedge \Box p \rightarrow q) \vee \Box(q \wedge \Box q \rightarrow p)$. We shall call the formula $\Box(p \wedge \Box p \rightarrow q) \vee \Box(q \wedge \Box q \rightarrow p)$ $r3$, and show that $\mathbf{K} \bigoplus r3$ has the finite model property.

Firstly, we define connected normal forms.

Definition 4.11 (Connected normal form)**.** A normal form α is **connected** if:

- $deg(\alpha) = 0$ or
- $\alpha > \beta_1$ & $\alpha > \beta_2$ & $\beta_1 \neq \beta_2 \Rightarrow (\beta_1 > \beta_2'$ or $\beta_2 > \beta_1')$ and
- $\forall \beta, \alpha > \beta \Rightarrow \beta$ is connected.

Now, we show that if a normal form is not connected, it is equivalent to $\bot$ in $\mathbf{K} \bigoplus r3$:

Assume that α is not connected; $\alpha > \beta_1$, $\alpha > \beta_2$, $\beta_1 \neq \beta_2$ and neither $\beta_1 > \beta_2'$ nor $\beta_2 > \beta_1'$.

Letting $p = \neg\beta_1$, and $q = \neg\beta_2$, under a simple substitution, the axiom $r3$ becomes $\Box(\neg\beta_1 \wedge \Box\neg\beta_1 \rightarrow \neg\beta_2) \vee \Box(\neg\beta_2 \wedge \Box\neg\beta_2 \rightarrow \neg\beta_1)$.

[2]This is similar to the construction used in [16] for the logic **D**.

Now, $\beta_1 \rightarrow \neg\beta_2$ and $\beta_2 \rightarrow \beta_2'$, so $\beta_1 \not> \beta_2'$ gives us $\beta_1 \rightarrow \neg\Diamond\beta_2$. Thus, we have $\beta_1 \rightarrow \neg\beta_2 \wedge \Box\neg\beta_2$, so, either $\beta_1 \rightarrow \neg\beta_1$, or $\Box(\neg\beta_1 \wedge \Box\neg\beta_1 \rightarrow \neg\beta_2)$.

Similar reasoning with β_2 leads us to conclude that either $\beta_1 \rightarrow \neg\beta_1$ or $\beta_2 \rightarrow \neg\beta_2$. Thus, we have a contradiction, and non-connected normal forms are equivalent to $\bot$ in $\mathbf{K} \bigoplus r3$.

Now, for a suitable model to realise a connected normal form α, we start with the graded model $\mathfrak{A}_\alpha$, and augment R_A by adding $\langle \gamma_1, \gamma_2 \rangle$ whenever $\gamma_1 > \gamma_2'$.

This model illustrates an important point; If $\alpha > \beta$, then $A \vDash \alpha$ as long as the A point sees *at least* one B point. Adding additional connections to other B points cannot create a situation where A is no longer true. And since $\beta \rightarrow \beta'$, our addition for connectedness is valid.

That is, if $\beta_1 > \gamma$, and $\beta_2' = \gamma$, it does not matter if the associated point B_1 is connected to C, B_2 or both, we will still have $B_1 \vDash \beta_1$.

$\mathbf{Alt_n}$

The modal condition $alt_n = \Box p_1 \vee \Box(p_1 \rightarrow p_2) \vee \ldots \vee \Box(p_1 \wedge \ldots \wedge p_n \rightarrow p_{n+1})$ places a bound on the number of points accessible from each point in a frame. That is, $\forall x, x_1 \ldots x_n(\bigwedge_{i=1}^{n+1} xRx_i \rightarrow \bigvee_{i \neq j} x_i = x_j)$. We define alt_n suitability:

Definition 4.12. A normal form α is alt_n **suitable** if:

- $deg(\alpha) = 0$ or
- If $\alpha > \beta_1, \alpha > \beta_2 \ldots, \alpha > \beta_{n+1}$ then $\exists i, j$ such that $i \neq j$ and $\beta_i = \beta_j$ and $\forall i, \beta_i$ is alt_n suitable.

It is essentially trivial to verify that if α is not alt_n-suitable then α is a contradiction in $\mathbf{K} \bigoplus alt_n$, and completely trivial to show that the frame for the graded model $\mathfrak{A}_\alpha$ will indeed verify alt_n. We include this example not for its interest, but for its application to the later, substantially more complex example of $\mathbf{KTB} \bigoplus alt_n$.

Symmetric logic

The symmetric modal logic $p \rightarrow \Box\Diamond p$ is an interesting condition because, intuitively, the way to do it is to have a form of the counter-correlate α°, that allows us to state $\alpha > \beta \rightarrow \beta^\circ > \alpha'$. See [32] for an example of attempting such an approach.

However, because there is no unique counter-correlate for any given β, this approach is unworkable. Instead, we define the **B**-suitable normal forms as:

Definition 4.13 (**B**-suitable)**.** A normal form α is **B-suitable** if:

- $deg(\alpha) = 0$ or $deg(\alpha) = 1$ or
- $\forall \beta, \alpha > \beta \Rightarrow \beta > \alpha''$ and β is **B** suitable.

Now it is clear that a normal form that is not **B**-suitable is equivalent to $\bot$ in **B**. Since $\alpha \rightarrow \alpha''$, and by the axiom of symmetry, $\alpha'' \rightarrow \Box\Diamond\alpha''$, we have $\alpha \rightarrow \Box\Diamond\alpha''$. If $\beta \not> \alpha''$, then $\beta \rightarrow \neg\Diamond\alpha''$. Thus, if $\alpha > \beta$, and $\beta \not> \alpha''$, we would have $\alpha \rightarrow \Box\Diamond\alpha''$ and $\alpha \rightarrow \neg\Box\Diamond\alpha''$, a contradiction.

To create a model realising a **B**-suitable normal form α, we take the symmetric closure of the graded model $\mathfrak{A}_\alpha$. Using similar reasoning to the case of connected models, if $\beta > \alpha''$ then adding (β, α) to R_α does not affect the truth of the statement $\langle \mathfrak{A}_A, B \rangle \vDash \beta$, and as long as that is unaffected, then for all points C, where $CR_\alpha B$, it is still the case that $\langle \mathfrak{A}_A, C \rangle \vDash \gamma$.

Thus, we may conclude that **B** has the finite model property.

Reflexive, Symmetric logic

While defining **L**-suitable forms for a particular logic can be easy, it is good to be able to combine various logics easily as well. A very simple example of this is the logic **KTB**, which combines reflexivity and transitivity. For defining **KTB**-suitable, we simply combine the definitions of **T**-suitable and **B**-suitable - A normal form is **KTB**-suitable if it is both **T**-suitable and **B**-suitable.

Deriving an appropriate model is again simply a matter of combining the model conditions of the two component logics - A **KTB**-suitable normal form α is satisfied in the symmetric, reflexive closure of the graded model $\mathfrak{A}_\alpha$.

$\mathbf{KTB} \bigoplus alt_n$

Having seen that it can be simple to combine logics to produce the FMP, it is important to note that the combining of known logics is not always as simple as the example of **KTB**. For an example of a more complex logical issue, take $\mathbf{KTB} \bigoplus alt_n$. All the components of the logic have already been explored, and as before, we can say that a normal form α is $\mathbf{KTB} \bigoplus alt_n$-suitable if it is **T**-suitable, **B**-suitable and alt_n-suitable.

However, when we attempt to take the reflexive/symmetric closure of a model, we add additional relations, causing the frame to fail the condition alt_n. This requires a more complex model construction.

To construct the relevant model, we start with the graded tree $\mathfrak{T}_\alpha$. Now, in the worst case for any world B, B sees n other worlds. By the definition of **KTB**-suitable, one of these other worlds B' is associated with the normal form β', and one of the worlds A'' is associated with the normal form form α'', where A, associated with α is the parent world of B.

Now, if we add the links (B, B), and (B, A), and simultaneously remove the links (B, B') and (B, A''), then we still satisfy the condition that $\langle \mathfrak{T}_\alpha, B\rangle \vDash \beta$, while not actually changing the number of alternative worlds accessible from B.

Thus, we can construct our appropriate model, and conclude that $\mathbf{KTB} \bigoplus alt_n$ has the FMP.

4.5.2 A failure

There are some cases where simple application of normal forms does not provide an easy way to show the finite model property. An example of this is attempting to generalise transitivity. According to [16], a normal form α is **K4** suitable if $\forall\beta_1, \gamma(\alpha > \beta_1 > \gamma \Rightarrow \exists\beta_2(\alpha > \beta_2, \beta_2' = \gamma, \beta_1 \succ \beta_2))$.

Now, the axiom $4 = \Box p \rightarrow \Box\Box p$ generalises neatly to $4_n = \bigwedge_{i=0}^{n} \Box^i p \rightarrow \Box^{n+1} p$ [10]. It is tempting to suppose that **K4**-suitability generalises in a similar fashion to $\mathbf{K4_n}$-suitability.

The obvious step to take is to generalise the $\succ$ relation to $\succ^n$, in much the same way $>$ was generalised to $>^n$. That is, $\alpha_1 \succ^n \alpha_2$ if $\alpha_2 >^n \beta \rightarrow \alpha_1 >^n \beta$. Just as ARB, in a transitive frame, implies $\alpha \succ \beta$ for the associated normal forms, in an n-transitive frame, ARB should imply $\alpha \succ^n \beta$. Then, a simple generalisation of **K4** suitability gives us:

Definition 4.14. A normal form α is $\mathbf{4_2}$**-suitable** if $\forall\beta_1, \gamma_1, \delta$, when $\alpha > \beta_1 > \gamma_1 > \delta$, either

- $\exists\beta_2$ such that $\alpha > \beta_2, \beta_2'' = \delta, \beta_1 \succ^2 \beta_2$ and $\gamma_1 \succ^2 \beta_2'$ or
- $\exists\gamma_2$ such that $\alpha >^2 \gamma_2, \gamma_2' = \delta, \beta_1' \succ^2 \gamma_2$ and $\gamma_1 \succ^2 \gamma_2$.

Of course, we add the usual caveats about the base case of $deg(\alpha) = 0$ always being $\mathbf{4_2}$-suitable, and that all of $\{\beta : \alpha > \beta\}$ must also be $\mathbf{4_2}$-suitable.

However, it turns out that this generalisation will not work; 4_n is too 'loose' a condition to properly dictate a frame. It is obvious that while the transitive closure of a frame is unique, the n transitive closure is not. See for instance Figure 4.4, which shows three different ways to provide the 2-transitive closure of

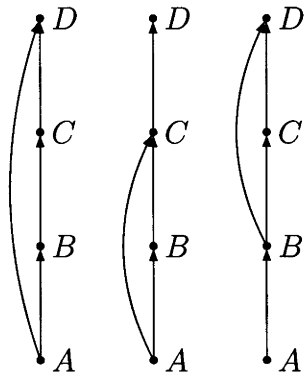

Figure 4.4: Non-unique closures

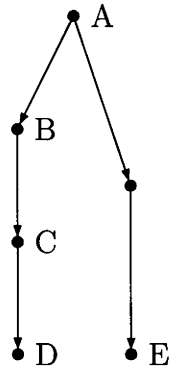

Figure 4.5: A failure in the ungraded frame

a simple 4 element chain. This looseness is reflected in the normal forms. While the $\succ$ condition can lead to a suitable frame for **K4** [16], it is not the case that $\succ^2$ constrains the normal form equally well.

To some extent, this is the same problem as arises when trying to define the counter-correlate - there isn't enough data to step back from α'' to α uniquely. In the case of 4, we are constraining the elements β such that ARB in the associated model, which we can do safely. In the case of 4_2, we must constrain the elements γ such that AR^2C in the associated model. This creates submodels of the ungraded model like Figure 4.5, where for the associated formulas γ and δ for the worlds D and E, we have $\gamma'' = \delta''$. Essentially, the inability of normal forms to see past a certain depth works against us. The normal form α, associated with A satisfies 4_2 suitability because A sees E, even though it actually needs to see D.

Another problem that arises is the situation of boundary cases at the very low degrees.

Take a set of leading terms $\{a, b, c, d, e\}$. From these we define the normal forms $\alpha = a \wedge \Diamond\{b\}$, $\beta = b \wedge \Diamond\{c\}$, $\gamma = c \wedge \Diamond\{d\}$ and $\delta = d \wedge \Diamond\emptyset$.

Then we have a set of normal forms $\varphi_1 \ldots \varphi_4$, where $\varphi_1 = e \wedge \Diamond\{\alpha\}$, $\varphi_2 = e \wedge \Diamond\{\beta\}$, $\varphi_3 = e \wedge \Diamond\{\gamma\}$ and $\varphi_4 = e \wedge \{\delta\}$.

Lastly, we define a normal form $\psi = e \wedge \Diamond\{\varphi_i : 1 \leq i \leq 4\}$.

If we associate these normal forms with various worlds, where φ_i is associated with a world P_i, and ψ is associated with a world O, we can produce the frame shown in Figure 4.6, where it can clearly be seen that while ψ is 4_2 suitable, the point A fails to satisfy 4_2. Thus, any definition of 4_2 suitability would have to refute the possibility of ψ being 4_2 suitable.

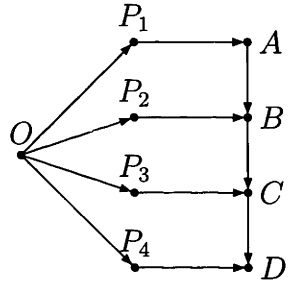

Figure 4.6: $\mathbf{K4_2}$-suitable, yet refuting 4_2

Chapter 5

Proof Search in S4

5.1 Introduction

Sequent calculi, or Gentzen systems, outlined in Chapter 2, can be used as a tool of automated reasoning. They provide a set of rules that are used to manipulate sets of formulas, called sequents. Instead of working with a large set of axioms and a couple of simple inference rules, a sequent calculus has but a couple of axioms, and many inference rules.

The strength of Gentzen systems for automated reasoning is that the rules can be applied backwards. If you desire to prove a formula φ, it is possible to start from φ and work backwards, saying φ is a theorem if one of $\phi_1, \ldots \phi_n$ is a theorem, and work from there until one of the required formulas is an axiom.

However, automated reasoning has a weakness, and that is the potential for a backwards derivation to loop infinitely. There are ways to deal with this problem, usually involving additional notation. This chapter will look at the causes of non-termination, and discuss two attempts to solve the problem. Heuerding's approach [24] is to add histories of what has already been done, while Pliuškevičius and Pliuškevičienė [36] attempt to solve the problem by adding various marks and indices to the syntax of the logic. We compare these two methods, and mention some flaws in the method of [36].

5.1.1 Termination issues in S4

The most basic Gentzen systems are extremely simple, requiring explicit structural rules for such simple operations as contraction (That is, a sequent with two occurrence of φ is identical to one with only a single occurrence of φ). For

purposes of this discussion, we shall ignore these simple calculi and restrict our attention to right-sided sequent calculi, leave the structural rules implicit, and assume that all formulas have been reduced to negation normal form, as defined in Chapter 2. This does not affect the correctness of the discussion, but the more powerful calculi are easier to work with. We shall discuss the sequent calculus **S4**. It is defined in Chapter 2, but for convenience, we reproduce it in Figure 5.1.

$$(\mathit{Axiom})\ \ p, \neg p, \Gamma \qquad\qquad (\mathit{Verum})\ \ \top, \Gamma$$

$$(\vee)\ \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma} \qquad\qquad (\wedge)\ \frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$$

$$(\Diamond)\ \frac{\Diamond\varphi, \varphi, \Gamma}{\Diamond\varphi, \Gamma} \qquad\qquad (\Box, \mathit{JUMP})\ \frac{\varphi, \Diamond\Delta}{\Box\varphi, \Diamond\Delta, \Gamma}$$

Figure 5.1: Right sided Gentzen calculus for **S4**

When used for automated proof search, this calculus is not guaranteed to terminate. The axiom, and the classical logical rules of $(\wedge)$ and $(\vee)$ pose no termination problems. However, the modal rules pose a significant problem for automated backwards proof search. The calculus, as presented, does not in general terminate. There are two problems that lead to non-termination.

The first problem with proof search in **S4** is looping via the rule $(\Diamond)$. This arises with a formula of the form $\Diamond p$. Backwards applying the rule $(\Diamond)$ to this formula produces the pair $\Diamond p, p$. The rule $(\Diamond)$ can be backwards applied to this pair to produce $\Diamond p, p, p$. And so ad infinitum.

The naive approach to resolving this problem would be to observe that unlike our other sequent rules, the rule $(\Diamond)$, when applied backwards, does not reduce the complexity of the sequent. A simple, but incorrect, approach to solving this problem would appear to be using an alternate $\Diamond$ rule, like this:

$$\frac{\varphi, \Gamma}{\Diamond\varphi, \Gamma}\ (\Diamond, \mathit{WRONG})$$

This clearly prevents the non-termination problem; After an application of $(\Diamond, \mathit{WRONG})$, the relevant formula is no longer in the premise, so the looping cannot occur. Unfortunately, it introduces a completeness problem. The formula $\Diamond(p \vee \Box\Diamond\neg p)$ is a theorem of **S4**, but as outlined in [24], if we attempt to derive

statements such as $\Diamond(p \vee \Box\Diamond\neg p)$ then we need the original rule $(\Diamond)$. Consider the difference:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{\Diamond(p \vee \Box\Diamond\neg p), p, \Box\Diamond\neg p, \Diamond\neg p, \neg p}(\vee)}{\Diamond(p \vee \Box\Diamond\neg p), p \vee \Box\Diamond\neg p, \Diamond\neg p, \neg p}(\Diamond)}{\Diamond(p \vee \Box\Diamond\neg p), \Diamond\neg p, \neg p}(\Diamond)}{\Diamond(p \vee \Box\Diamond\neg p), \Diamond\neg p}(\Box, \mathit{JUMP})}{\Diamond(p \vee \Box\Diamond\neg p), p, \Box\Diamond\neg p}(\vee)}{\Diamond(p \vee \Box\Diamond\neg p), p \vee \Box\Diamond\neg p}(\Diamond)}{\Diamond(p \vee \Box\Diamond\neg p)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg p}{\Diamond\neg p}(\Diamond, \mathit{WRONG})}{p, \Box\Diamond\neg p}(\Box, \mathit{JUMP})}{p \vee \Box\Diamond\neg p}(\vee)}{\Diamond(p \vee \Box\Diamond\neg p)}(\Diamond, \mathit{WRONG})$$

The left shows the derivation of the formula using the rule $(\Diamond)$, while the right shows the failed attempt at backwards derivation using the rule $(\Diamond, \mathit{WRONG})$. As can be seen, having the rule $(\Diamond)$ preserve the original formula in some form is essential to deriving some formulas. However, we have already established that repeated application of $(\Diamond)$ can cause an infinite loop, which is problematic for automated application.

The better way to solve the problem is simply to ensure that $(\Diamond)$ is not applied repeatedly to a particular formula. There are a couple of ways of doing this, however, all are basically isomorphic. We shall look at two methods, both of which boil down to sequestering the $\Diamond\varphi$ formulas after application of the $(\Diamond)$ rule.

The second problem is an infinite looping branch, as with formulas of the form $\Diamond\Box\varphi$. If we establish rules to limit the repeated application of $(\Diamond)$ to a given formula, it must still be possible to apply $(\Diamond)$ more than once, as the issues with $(\Diamond, \mathit{WRONG})$ shows. In particular, we need to be able to apply the $(\Diamond)$ rule after using the $(\Box, \mathit{JUMP})$ rule. However, this requires further limits to prevent a second kind of infinite loop, as shown below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Box p, p}(\Diamond)}{\Diamond\Box p, \Box p, p}(\Box, \mathit{JUMP})}{\Diamond\Box p, p}(\Diamond)}{\Diamond\Box p, \Box p}(\Box, \mathit{JUMP})}{\Diamond\Box p}(\Diamond)$$

5.2 Heuerding's Calculus

$$(Axiom)\ \ H\|\Sigma|p,\neg p,\Gamma \qquad\qquad (Verum)\ \ H\|\Sigma|\top,\Gamma$$

$$(\vee)\ \frac{H\|\Sigma|\psi,\chi,\Gamma}{H\|\Sigma|\psi\vee\chi,\Gamma} \qquad\qquad (\wedge)\ \frac{H\|\Sigma|\psi,\Gamma \qquad H\|\Sigma|\chi,\Gamma}{H\|\Sigma|\psi\wedge\chi,\Gamma}$$

$$(\Diamond, NEW)\ \frac{\varepsilon\|\Diamond\varphi,\Sigma|\varphi,\Gamma}{H\|\Sigma|\Diamond\varphi,\Gamma}\ \Diamond\varphi\notin\Sigma \qquad\qquad (\Diamond)\ \frac{H\|\Sigma|\varphi,\Gamma}{H\|\Sigma|\Diamond\varphi,\Gamma}\ \Diamond\varphi\in\Sigma$$

$$(\Box, JUMP)\ \frac{\varphi,\Delta,H\|\Sigma|\varphi,\Sigma}{H\|\Sigma|\Box\varphi,\Box\Delta,\Gamma}\ \varphi\notin H$$

Figure 5.2: Heuerding's calculus for **S4**

Heuerding's calculus [23], shown in full in Figure 5.2 deals with the nontermination problems of **S4** by making sequents consist of three sets of formulae. There is the primary set, to which the logical rules apply as usual. There are also formula sets Σ and H, separated from our main formula by $\|$ and $|$. Thus, where an ordinary right-handed sequent contains only a single formula set Γ, a sequent in this calculus has the form $H\|\Sigma|\Gamma$.

These two additional formula sets provide a history of our rule applications, enabling us to avoid infinite loops in backwards derivation. In a backwards derivation, both sets start empty, and application of our modal rules adds formulas to these history sets. If a set is already present in the history, then this blocks the application of the relevant rule, so we cannot start the infinite loop.

For an example of a derivation in this new calculus, consider the previously mentioned example of $\Diamond(p\vee\Box\Diamond\neg p)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{\varepsilon\|\Diamond(p\vee\Box\Diamond\neg p),\Diamond\neg p|\neg p,p,\Box\Diamond\neg p}(\vee)}{\varepsilon\|\Diamond(p\vee\Box\Diamond\neg p),\Diamond\neg p|\neg p,p\vee\Box\Diamond\neg p}(\Diamond)}{\varepsilon\|\Diamond(p\vee\Box\Diamond\neg p),\Diamond\neg p|\neg p,\Diamond(p\vee\Box\Diamond\neg p)}(\Diamond, NEW)}{\Diamond\neg p\|\Diamond(p\vee\Box\Diamond\neg p)|\Diamond\neg p,\Diamond(p\vee\Box\Diamond\neg p)}(\Box, JUMP)}{\varepsilon\|\Diamond(p\vee\Box\Diamond\neg p)|p,\Box\Diamond\neg p}(\vee)}{\varepsilon\|\Diamond(p\vee\Box\Diamond\neg p)|p\vee\Box\Diamond\neg p}(\Diamond, NEW)}{\varepsilon\|\varepsilon|\Diamond(p\vee\Box\Diamond\neg p)}$$

The non modal rules of the calculus are essentially unchanged. They do not modify the additional formula sets Σ and H, and their application is not modified

by the contents of these sets. They act only on the primary formula set. The only way to interact with the sets Σ and H is via the new modal rules.

When we write the new modal rules, as in Figure 5.2, then we need to add a new element to the rules. Before, we had the premise (top line), the conclusion (bottom line) and the rule name (to the left). Now, we add conditionals, written to the right of the rules. If the conditional is not fulfilled, then the rule may not be used. This prevents the repeated application of $(\Box, \mathit{JUMP})$ and $(\Diamond, \mathit{NEW})$.

The set Σ is used to prevent the simpler infinite loop problem, where we simply repeat the $(\Diamond)$ rule without limit. In the basic calculus, backwards applying the $(\Diamond)$ rule to a formula $\Diamond\varphi$ produces a sequent containing both $\Diamond\varphi$ and φ. In our new calculus, we have two rules for $\Diamond$. When we apply the $(\Diamond, \mathit{NEW})$ rule, we place the formula φ into the primary formula set, but $\Diamond\varphi$ into the set Σ. Since the $\Diamond$ rules are not applicable to a formula in Σ, they cannot loop, but $\Diamond\varphi$ will still be available when it is needed, unlike in the naive example of $(\Diamond, \mathit{WRONG})$.

The rule $(\Diamond)$ also produces the formula φ, but does not add $\Diamond\varphi$ to Σ. This is because its side condition requires that $\Diamond\varphi$ is already in Σ, and adding an additional instance of $\Diamond\varphi$ to Σ is not necessary. The other difference is that the rule $(\Diamond, \mathit{NEW})$ resets the formula set H. We shall discuss the importance of that shortly.

The set H is necessary to prevent the more complex infinite loop, caused by such formulas as $\Diamond\Box p$, since the side condition of the rule $(\Box, \mathit{JUMP})$ will prevent repeated application of the rule. Thus, when attempting a backward derivation of $\Diamond\Box p$, this calculus will terminate:

$$\dfrac{\dfrac{\dfrac{p \| \Diamond\Box p | p, \Box p}{p \| \Diamond\Box p | p, \Diamond\Box p}\,(\Diamond)}{\varepsilon \| \Diamond\Box p | \Box p}\,(\Box, \mathit{JUMP})}{\varepsilon \| \varepsilon | \Diamond\Box p}\,(\Diamond, \mathit{NEW})$$

Here, because the set H already contains the formula p, we cannot apply the rule $(\Box, \mathit{JUMP})$ to $\Box p$, and our attempted derivation fails.

The subtlety that arises in this calculus is the interaction between the rules for $(\Diamond)$ and $(\Box, \mathit{JUMP})$. The rule $(\Box, \mathit{JUMP})$, when applied, copies the formulas in Σ back to the primary sequent. This ensures that they are still available for use, as in our example of deriving $\Diamond(p \vee \Box\Diamond\neg p)$. When the $(\Diamond)$ rule is applied to a formula $\Diamond\varphi$, and $\Diamond\varphi$ is not already in the set Σ, we reset the history H to the empty set. The change in Σ changes the state of our derivation enough to render our prior history irrelevant. However, since we are working with finite formulas,

and Σ never resets, these interactions can only occur a finite number of times, and will not prevent termination.

It is necessary to provide the reset of histories, otherwise the calculus would become incomplete. Consider the derivation of the formula $\Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p)$:

$$
\begin{array}{cl}
\text{Axiom} & \\ \hline
\varepsilon \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p, \Diamond\neg p | \neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), p & \\ \hline
\varepsilon \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p, \Diamond\neg p | \neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p & (\Diamond) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p, \Diamond p, \Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p | \Diamond\neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p & (\Diamond, \mathit{NEW}) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p | \Box\Diamond p, \Box\Diamond\neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p & (\Box, \mathit{JUMP}) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p | \Box\Diamond p \vee \Box\Diamond\neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p & (\vee) \\ \hline
\varepsilon \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p | p, \Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Box, \mathit{JUMP}) \Leftarrow \\ \hline
\varepsilon \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p), \Diamond p, | p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Diamond) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p, \Diamond p, \Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) | \Diamond p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Diamond, \mathit{NEW}) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) | \Box\Diamond p, \Box\Diamond\neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Box, \mathit{JUMP}) \\ \hline
\Box\Diamond p \vee \Box\Diamond\neg p \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) | \Box\Diamond p \vee \Box\Diamond\neg p, \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\vee) \\ \hline
\varepsilon \| \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) | \Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Box, \mathit{JUMP}) \\ \hline
\varepsilon | \Diamond\Box(\Box\Diamond p \vee \Box\Diamond\neg p) & (\Diamond, \mathit{NEW})
\end{array}
$$

If we did not reset the history H after modifying Σ, then the backwards application of $(\Box, \mathit{JUMP})$ marked with $\Leftarrow$ in the derivation above would not be possible, and the derivation would fail.

Another subtlety of the $(\Box, \mathit{JUMP})$ rule is that it adds all $\Box$ formulas in the primary formula set Γ to H, instead of only the primary formula of the rule.

This is actually an optimisation, to limit the amount of backtracking needed. Consider that we must allow for backtracking after applying the rule $(\Box, \mathit{JUMP})$. Suppose we apply the rule $(\Box, \mathit{JUMP})$ to a formula set containing both $\Box\varphi, \Box\chi, \Gamma$, producing the new primary formula set φ, Σ. If we later derive a sequent $\Box\chi, \Gamma'$, without modifying the set Σ, then applying the rule $(\Box, \mathit{JUMP})$ will produce a primary formula set χ, Σ, which we could have produced directly from our original formula set.

If the set Σ is changed, then we empty our history anyway. If our derivation from φ, Σ had failed, we'd need to backtrack and check the results of applying the rule $(\Box, \mathit{JUMP})$ to the primary sequent. There's no need to check this twice, so adding χ to our history H saves us some work.

For proof that this calculus is sound, complete and terminating, see [23] or [24].

5.2.1 Refining Heuerding's calculus

As the astute reader will have noticed, the primary purpose of the rule $(\Diamond)$ is to act on $\Diamond\varphi$ formulas that are already in Σ. Now, if we have not invoked the rule $(\Box, JUMP)$ since $\Diamond\varphi$ was added to Σ, this is clearly unnecessary, as the formula φ was added to our primary sequent when we added $\Diamond\varphi$ to Σ, and we won't need to add it twice.

On the other hand, when we jump, then we copy all the formulas $\Diamond\varphi$ in Σ back to our primary sequent, and promptly remove them again with the rule $(\Diamond)$. This action is almost automatic, and certainly does not enhance our understanding of the derivation in any meaningful way.

As an alternative, we know that all formulas in Σ have the form $\Diamond\varphi$ for some φ. Suppose we allow the $\Diamond$ to be implicit rather than making it explicit every time. That is, when we apply the $(\Diamond, NEW)$ rule, instead of adding $\Diamond\varphi$ to Σ, we add φ.

If we do this, then we do not need to apply the rule $(\Diamond)$ after a jump, since the formulas we copy across from Σ will already have the $\Diamond$ removed. Indeed, we should never need to apply the rule $(\Diamond)$ to a formula $\Diamond\varphi$, as whenever φ is in Σ, then we will already have added φ to our primary sequent, either by using the rule $(\Diamond, NEW)$ or by copying it back from Σ with the rule $(\Box, JUMP)$. As such, we may remove the rule, $(\Diamond)$ from our calculus, producing the revised set of modal rules in Figure 5.3.

$$(\Diamond, NEW)\ \frac{\varepsilon \| \varphi, \Sigma | \varphi, \Gamma}{H \| \Sigma | \Diamond\varphi, \Gamma}\ \varphi \notin \Sigma \qquad (\Box, JUMP)\ \frac{\varphi, \Delta, H \| \Sigma | \varphi, \Sigma}{H \| \Sigma | \Box\varphi, \Box\Delta, \Gamma}\ \varphi \notin H$$

Figure 5.3: Revised modal rules for history-based **S4**

5.3 The Marks and Indices method

Heuerding's calculus, based on maintaining history sets, shows the normal approach used to guarantee termination for proof search in **S4** and other logics. However, [36] presents a different calculus. Instead of maintaining a extra formula sets, this calculus is based on marking the various modalities to indicate usage. The claim is made that this enables us to eliminate the added overhead of a history set, and restrict the need to backtrack after choosing to apply the $(\Box, JUMP)$ rule to the wrong formula.

$$(\Diamond^*)\ \frac{\varphi, \Diamond^*\varphi, \Gamma}{\Diamond\varphi, \Gamma} \qquad (\Box^{\sigma+})\ \frac{\Gamma^{\sigma+}, \Diamond^*\Gamma^{\sigma+}, \varphi}{\Sigma, \Diamond^*\Gamma, \Box^{\sigma}\varphi}\ (\Box^{i})\ \frac{\Gamma^{i+}, \Diamond^*\Gamma, \varphi}{\Sigma, \Diamond^*\Gamma, \Box^{i}\varphi} \qquad (\Box)\ \frac{\Gamma, \Diamond^*\Gamma, \varphi}{\Sigma, \Diamond^*\Gamma, \Box\varphi}$$

Figure 5.4: The marks and indices rules for **S4**

There are two marks used by this method, $+$ and $*$, and two sets of indices, i and $\circ j$.

The marks and indices method is a two step process. First, indices are assigned to all the $\Box$ subformulas of the sequent, like so:

Firstly, $\Box$ formulas that are not within the scope of a $\Diamond$ are left alone. In a backwards derivation, only formulas within a $\Diamond$ formula reoccur, so we do not need to mark them to prevent looping.

Secondly, every $\Box$ within the scope of a $\Diamond$ receives a unique index i. This index is either a simple index, written $\Box^i$ or a special index, written $\Box^{\circ i}$.

An index defaults to being simple. $\Box\varphi$ receives a special index only if:

- φ contains a subformula of the form $\Diamond\psi$ and
- φ has no subformula $\Box\chi$, such that χ has a subformula of the form $\Diamond\phi$.

Then we modify our modal rules that convert the indices to marks when applied. In these sequents, $\Gamma^{\sigma+}$ means converting all instances of the index σ (which may have the form i or $\circ i$) to the mark $+$. The basic logical rules of $(\wedge)$ and $(\vee)$, and the axiom, remain as outlined in Figure 5.1. The new modal rules are shown in Figure 5.4

These rules are backward applied subject to a particular strategy - The intent is to limit the amount of backtracking needed. The strategy hinges on ordering your application of modal rules as follows:

1. Apply the rule $(\Diamond^*)$ if possible.
2. Apply the rule $(\Box)$ if possible.
3. Apply the rule $(\Box^{\sigma+})$ if possible. This rule is always applicable to formulas with a special index of the form $\Box^{\circ k}$. It is only applicable to formulas of the form $\Box^i A$ if the following conditions are met:
   - In $\Diamond^*\Gamma$, there is an occurrence of $\Box^i\varphi$ where all subformulas of φ are marked. That is, if $\Box^\mu\psi$ is a subformula of A, then $\mu = +$.
   - $\Diamond^*\Gamma$ contains no subformulas of the form $\Box^{\circ k}\psi$ for any k.

4. apply the rule $\Box^i$ to formulas of the form $\Box^i\varphi$, subject to the following rules of priority:

   - Formulas containing some $\Box^{\circ k}\psi$ as a subformula
   - Formulas $\Box^i\varphi$ that are subformulas of some other $\Box^\mu\psi$ in the sequent. If there are multiple such formulas then we give priority to the one that is a subformula of as many distinct $\Box^\mu\psi$ formulas as possible.
   - Other formulas of the form $\Box^i\varphi$.

For example, consider again the formula $\Diamond(p \vee \Box\Diamond\neg p)$. We add indices to produce the formula $\Diamond(p \vee (\Box^{\circ 1}\Diamond\neg p)$, and then run our derivation process

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{\Diamond^*(p \vee \Box^+\Diamond\neg p), \Diamond^*\neg p, \neg p, p, \Box^+\Diamond\neg p}(\vee)}{\Diamond^*(p \vee \Box^+\Diamond\neg p), \Diamond^*\neg p, \neg p, p \vee \Box^+\Diamond\neg p}(\Diamond^*)}{\Diamond^*(p \vee \Box^+\Diamond\neg p), \Diamond\neg p, p \vee \Box^+\Diamond\neg p}(\Box^{\circ 1+})}{\Diamond^*(p \vee \Box^{\circ 1}\Diamond\neg p), p, \Box^{\circ 1}\Diamond\neg p}(\vee)}{\Diamond^*(p \vee \Box^{\circ 1}\Diamond\neg p), p \vee \Box^{\circ 1}\Diamond\neg p}(\Diamond^*)}{\Diamond(p \vee \Box^{\circ 1}\Diamond\neg p)}$$

5.3.1 Method discussion

For backwards proof search, the $*$ marking on $\Diamond$ performs the same purpose as the Σ set in Heureding's method: It prevents repeated application of the $(\Diamond)$ rule to a formula. Since the rule $(\Diamond^*)$ can only be invoked on unmarked $\Diamond$ formulas, each application of the rule must be to a different formula.

The $+$ marking performs some of the same functions as the histories in Heureding's method. However, Heureding's method always adds a formula to the history set H when the $(\Box, \mathit{JUMP})$ rule is invoked, and resets the history occasionally. The marks and indicies method instead refrains from marking a formula if there is a chance we shall need to use the formula again, and never resets marks. This in turn requires multiple rules for handling formulas of the form $\Box\varphi$, depending on how we plan to handle the mark.

Consider the formula $\Diamond\Box p$, the formula that previously demonstrated the nontermination problems arising from transitivity. When we apply the marks and indices method, we mark the formula to $\Diamond\Box^1 p$, and then produce the following terminating derivation:

$$\dfrac{\dfrac{\Diamond^*\Box^+p, \Box^+p, p}{\Diamond^*\Box^1p, \Box^1p}\,(\Box^{1+})}{\Diamond\Box^1p}\,(\Diamond^*)$$

Thus, this calculus clearly provides termination in circumstances where the basic sequent calculus does not. However, as we shall soon see, termination in some simple cases is not the same as a proper proof of termination in all cases.

We have three transitivity rules, and our choice of rule depends on how we intend to manipulate the marks. The first rule, $(\Box^{\sigma+})$ introduces a mark on all instances of a particular index σ, effectively removing the subformula $\Box^\sigma\varphi$ from consideration in future. The second rule only marks a couple of instances of an index. It prevents us from applying a transitivity rule to a particular $\Box^i\varphi$ twice in a row, but does not rule out reusing $\Box^i\varphi$ higher in our derivation.

The third transitivity rule exists only for unindexed boxes: occurring outside the scope of any $\Diamond$ formula, they will not reoccur once the rule has been applied, and there is no need to mark them in our history.

The definition of strongly special formulas $\Box^{\circ k}$, and consequent restriction on the rule $(\Box^{\sigma+})$, is necessary to preserve completeness. Consider, for example, a formula of the form $\Diamond(\Box p \vee \Box\Box\Diamond\neg p)$. If we apply the rule $(\Box^{\sigma+})$ to the subformula $\Box p$ too early, then it is locked out of consideration, and the attempted derivation fails, as shown below:[1]

$$\begin{array}{c l}
\text{Axiom} & \\ \hline
\Diamond^*(\Box^+p \vee \Box^2\Box^+\Diamond\neg p), \Diamond^*\neg p, \neg p, p, \Box^+p \vee \Box^2\Box^+\Diamond\neg p & \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^+\Diamond\neg p), \Diamond^*\neg p, \neg p, \Box^1p, \Box^2\Box^+\Diamond\neg p & (\Box^{\circ 1}) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^+\Diamond\neg p), \Diamond\neg p, \Box^1p, \Box^2\Box^+\Diamond\neg p & (\Diamond^*) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^+\Diamond\neg p), \Diamond\neg p, \Box^1p \vee \Box^2\Box^+\Diamond\neg p & (\vee) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), \Box^{\circ 3}\Diamond\neg p, \Box^1p, \Box^+\Box^{\circ 3}\Diamond\neg p & (\Box^{\circ 3+}) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), \Box^{\circ 3}\Diamond\neg p, \Box^1p \vee \Box^+\Box^{\circ 3}\Diamond\neg p & (\vee) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), p, \Box^+p, \Box^2\Box^{\circ 3}\Diamond\neg p & (\Box^2) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), p, \Box^+p \vee \Box^2\Box^{\circ 3}\Diamond\neg p & (\vee) \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), \Box^1p, \Box^2\Box^{\circ 3}\Diamond\neg p & (\Box^1) \Leftarrow \\ \hline
\Diamond^*(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p), \Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p & (\vee) \\ \hline
\Diamond(\Box^1p \vee \Box^2\Box^{\circ 3}\Diamond\neg p) & (\Diamond^*)
\end{array}$$

[1]Of course, in **S4**, we can apply modal reduction principles to reduce this and several later examples, producing a simpler formula that lacks the properties we want. It's not hard to add complexity to the examples to prevent this, and we want to keep the examples to the simplest possible demonstration of a property

If, at the step marked with $\Leftarrow$, we use the rule $\Box^{1+}$ instead of the rule $\Box^1$, then the subformula $\Box^1 p$ is locked out of consideration, and is not available for the final application of the rule $\Box^{1+}$. Thus, the derivation would fail if we were allowed to apply $\Box^{\sigma+}$ to $\Box^i$ formulas while there is a formula of the form $\Box^{\circ k}\psi$ still available in the sequent.

5.3.2 Issues

As is immediately apparent, there are some drawbacks to this method. The indexing process adds some complexity, especially with the distinction between simple and special indices. Secondly, more importantly, we must check for subformulas, while the traditional method only needs to check for equality.

The need to check for subformulas is not necessarily insoluble. During the marking pass, we can do all the processing needed to create a list of subformulas, making it quick to check if one formula is a subformula of another. Since the subformula checking is only necessary on indexed formulas, it suffices to augment the indices i to indices i, j such that if a formula has indices a, b, where $a > i$, $b < j$, then $\Box^{a,b}\varphi$ is a subformula of $\Box^{i,j}\psi$[2].

So, for example, if we mark the formula $\Diamond(\Box p \vee \Box(\Box p \wedge \Box\Diamond p)$ with these augmented indices, we'd get the marked formula $\Diamond(\Box^{1,1}p \vee \Box^{2,4}(\Box^{3,3}p \wedge \Box^{4,4}\Diamond p)$, making it easy to quickly check for subformulas. That is, we can quickly see that $\Box^{3,3}p$ and $\Box^{4,4}\Diamond p$ are subformulas of $\Box^{2,4}\varphi$, while $\Box^{1,1}p$ is not.[3]

Termination, lack thereof

The most significant problem of the calculus is that it is non-terminating, contrary to the claim of termination in [36] Since the sole motivation for using anything but the basic calculus for **S4** is to provide termination, this is something of a problem.

Consider as a first example, the formula $\Diamond(\Box\Box\Box p \wedge \Box\Box\Box p)$, and observe the reduction tree (for simplicity's sake, we omit most of the branching for $(\wedge)$, focusing only on a single non-terminating branch). Let φ be the formula $\Box^1\Box^2\Box^3 p \wedge \Box^4\Box^5\Box^6 p$.

[2]If we view formulas as trees, this prevents a tail recursive indexing method. However, that ship sailed once we started caring about the distinction between regular and special indices.

[3]Technically, $\Box p$ is a subformula of $\Box(\Box p \wedge \Box\Diamond p)$. However, in the marks and indices method, $\Box^i\varphi$ and $\Box^j\varphi$ are different unless $i = j$. This confusion is a drawback of the method.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Diamond^*\varphi, \Box^1\Box^2\Box^3 p, \Box^5\Box^6 p}(\Box^1)}{\Diamond^*\varphi, \Box^1\Box^2\Box^3 p \wedge \Box^+\Box^5\Box^6 p, \Box^5\Box^6 p}\ \text{branch omitted}\ (\wedge)}{\Diamond^*\varphi, \Box^4\Box^5\Box^6 p, \Box^2\Box^3 p}\Box^4}{\Diamond^*\varphi, \Box^+\Box^2\Box^3 p \wedge \Box^4\Box^5\Box^6 p, \Box^2\Box^3 p}\ \text{branch omitted}\ (\wedge)}{\Diamond^*\varphi, \Box^1\Box^2\Box^3 p}(\Box^1)}{\Diamond^*(\varphi), \varphi}(\wedge)}{\Diamond(\Box^1\Box^2\Box^3 p \wedge \Box^4\Box^5\Box^6 p)}(\Diamond^*)$$

That's a simple example, and could perhaps be solved by extending the subformula checking, though as mentioned before, subformula checking is hard to implement. We were lucky that the limited subformula checking could be handled so easily.

There is another similar, more significant non-termination example, arising from the condition on applying $(\Box^{\sigma+})$ to simply indexed formulas:

The formula for consideration has the form $\Diamond((\Box p \vee \Box p) \wedge \Box\Diamond p)$. Let φ be the formula $(\Box^1 p \vee \Box^2 p) \wedge \Box^{\circ 3}\Diamond p)$, in the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Diamond^*(\varphi), p, (\Box^+ p \vee \Box^2 p) \wedge \Box^{\circ 3}\Diamond p)}(\Box^1)}{\Diamond^*(\varphi), p, \Box^1 p \vee \Box^+ p}\ \text{branch omitted}\ (\wedge)}{\Diamond^*(\varphi), p, (\Box^1 p \vee \Box^+ p) \wedge \Box^{\circ 3}\Diamond p)}(\Box^2)}{\Diamond^*(\varphi), p, \Box^+ p, \Box^2 p}(\vee)}{\Diamond^*(\varphi), p, \Box^+ p \vee \Box^2 p}\ \text{branch omitted}\ (\wedge)}{\Diamond^*(\varphi), p, (\Box^+ p \vee \Box^2 p) \wedge \Box^{\circ 3}\Diamond p)}(\Box^1)}{\Diamond^*(\varphi), \Box^1 p, \Box^2 p}(\vee)}{\Diamond^*(\varphi), \Box^1 p \vee \Box^2 p}\ \text{branch omitt}$$

$$\dfrac{\Diamond^*(\varphi), \varphi}{\Diamond(\varphi)}(\Diamond^*)$$

This example might be fixable as well. Maybe. In the meantime, we must abandon the marks and indices method as a convoluted failure.

However, such a fix comes at the cost of the simplifications provided by using right-sided sequents and negation normal form, so is unlikely to prove useful for automated reasoning.

5.4 Alternative Marking Method

Considering the problems that arise with the marks and indices method, it is tempting to assume that any method based on marking formulas instead of histories is doomed to failure.

However, recall that the rule ($\Diamond^*$) is essentially identical to the Σ set of the histories method. Indeed, from a theoretical viewpoint, there is little difference between using marks carried around by the formula, and using a history as a lookup table. A little thought will reveal that it is possible to produce a calculus where the marking rules behave essentially identically to the H sets of the histories method, like so:

Firstly, we assume that all modalities, both $\Box$ and $\Diamond$ have been assigned a unique index i. We also have two marked modalities, $\Diamond^*$ and $\Box^+$ which are part of the syntax.

Secondly, we have two operations; Firstly, we can mark indices taking a modality $\Box^i$ to $\Box^{i+}$ and $\Diamond^i$ to $\Diamond^{i*}$. We represent this operation with Γ^{i*} or Γ^{i+} to represent marking all instances of the index i within Γ with the mark $*$ or $+$. Secondly, we can reset a mark, reversing the operation. We represent this with the operation Γ°, which replaces all $+$ marks within Γ with the original indices.[4]

Lastly, we have the following modal rules:

$$(\Diamond^i)\ \frac{\Diamond^* A, A, (\Gamma^\circ)^{i*}}{\Diamond^i A, \Gamma}$$

$$(\Box^i)\ \frac{A, (\Sigma^{I+})^{i+}, \Diamond^*(\Sigma^{I+})^{i+}}{\Box^i A, \Box^I \Delta, \Diamond^* \Sigma, \Gamma}$$

As can be seen by comparison to the histories method, marking a $\Diamond$ formula is equivalent to adding it to the Σ set, and marking a $\Box$ formula is roughly equivalent to adding it to the H set

Now, this method is not perfectly isomorphic to the histories method. It's handling of repeated subformulas is different: compare $\Diamond(\Box\varphi \vee \Box\varphi)$, for an extremely simple example. Again, while too simple an example could be trivially resolved during preprocessing, there can be more complicated examples.

[4]It's not necessary to reproduce the original indices, just so long as they maintain the uniqueness condition on the original indices. For instance, it would be possible to simply keep a counter, incremented when an index was assigned, and use values from the counter to guarantee the use of a completely fresh index.

In the histories method, our derivation adds $\Box\varphi$ to the history no matter which side of the $\vee$ we resolve first, and prevents us from resolving the second. This is arguably the correct behaviour.

On the other hand, the index based method assigns each modality an index, producing $\Diamond^1(\Box^2\varphi \vee \Box^3\varphi)$. The method views $\Box^2\varphi$ and $\Box^3\varphi$ as being different formulas, and will attempt to resolve them separately. While this behaviour is unfortunate, it is a necessary consequence of mark based methods. The computation time that would be consumed by equality checking during the marking process outweighs the consequences of this edge case.

Nonetheless, this indexed method is sufficiently close to the histories method that we may show soundness and completeness by recourse to the soundness and completeness of the history method. The soundness of the method is trivial, since any derivation in this marked method can be replicated exactly in the basic **S4** sequent calculus, just by stripping out the indices.

The completeness of this method is also simple. Any derivation admissible in the history method can be reproduced in this calculus. Showing this simply requires comparing the circumstances under which a formula is marked and the circumstances under which the history method would transfer a formula to one of its sets Σ or H. If a derivation in the history method would require use of the $(\Box, \mathit{JUMP})$ rule on a formula $\Box\varphi$, then φ would not be in H, and in the mark method, the formula $\Box^i\varphi$ would not be marked. Likewise, if the history method would require use of $(\Diamond)$ or $(\Diamond, \mathit{NEW})$ on a formula $\Diamond\varphi$, then the corresponding formula $\Diamond^i$ will not be marked.

Showing termination is a matter of showing steady reduction in complexity. In the previous method could have an increase in the number of unmarked boxes when the weak transitivity rule was applied. However, in this calculus, a $\Diamond$ formula, once marked, is never unmarked. Further, the $\Box$ formulae are only unmarked when a $\Diamond$ formula is marked. Thus, in a sequent with n $\Diamond$ formulae and m $\Box$ formulae, the rule $(\Box^i)$ can be applied at most m times before applying the rule $(\Diamond^i)$. Since the rule $(\Diamond^i)$ applies only once to each $\Diamond$ formula, it can be easily seen that the modal rules may be applied to this sequent at most $n * m$ times.

However, the method is so close to the history based method that it raises the question of what benefit can be gained from adopting this method.

Certainly, the method provided has no purpose but to demonstrate that a marking based calculus may actually be viable[5]. However, it appears self-evident

[5]However, the viability of a calculus that never resets its marks, like the prior calculus,

that pre-processing the sequent before commencing reasoning has some benefits.

As a simple example, $\Box$ formulae outside the scope of $\Diamond$ formulae need not be added to the history. The marking provides a way to keep track of this, and reduces the size of the history that needs to be kept. Thus, marking provides a more efficient way of deriving formulas such as $\Box(p \vee \Box(q \vee \Box(p \vee \neg p)))$, as the history method would end with a history set $\{\Box(p \vee \Box(q \vee \Box(p \vee \neg p))), \Box(q \vee \Box(p \vee \neg p))), \Box(p \vee \neg p)))\}$, which is wasted overhead in the derivation process. However, such overhead appears to be generally small, especially compared to the cost of having two distinct $\Box$ rules and operators.

Further, the strategy of reducing $\Box$ outside $\Diamond$ formulas first is sound, as formulas within the scope of a $\Diamond$ will be brought back by the jump rule. Once the derivation is underway, it is much harder to determine the original state of a formula without marking. Consider the formulas $\Box(p \vee \neg p) \vee \Diamond\Box q$ and $\Box q \vee \Diamond\Box(p \vee \neg p)$:

With the strategy of reducing a $\Box$ outside a $\Diamond$ before one within[6]:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{p, \neg p, \Diamond\Box q}\vee}{p \vee \neg p, \Diamond\Box q}(\Box, \mathit{JUMP})}{\Box(p \vee \neg p), \Diamond\Box q, \Box q}(\Diamond)}{\Box(p \vee \neg p), \Diamond\Box q}(\vee)}{\Box(p \vee \neg p) \vee \Diamond\Box q} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{\Diamond\Box(p \vee \neg p), p, \neg p}(\vee)}{\Diamond\Box(p \vee \neg p), p \vee \neg p}(\Box, \mathit{JUMP})}{q, \Diamond\Box(p \vee \neg p), \Box(p \vee \neg p)}(\Diamond)}{q, \Diamond\Box(p \vee \neg p)}(\Box, \mathit{JUMP})}{\Box q, \Diamond\Box(p \vee \neg p), \Box(p \vee \neg p)}(\Diamond)}{\Box q, \Diamond\Box(p \vee \neg p)}(\vee)}{\Box q \vee \Diamond\Box(p \vee \neg p)}$$

On the other hand, compare the exact opposite strategy, with reducing a $\Box$ inside the $\Diamond$ before one without:

$$\dfrac{\dfrac{\dfrac{\Diamond\Box q, q}{\Box(p \vee \neg p), \Diamond\Box q, \Box q}(\Box, \mathit{JUMP})}{\Box(p \vee \neg p), \Diamond\Box q}(\Diamond)}{\Box(p \vee \neg p) \vee \Diamond\Box q}(\vee) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Axiom}}{\Diamond\Box(p \vee \neg p), p, \neg p}(\vee)}{\Diamond\Box(p \vee \neg p), p \vee \neg p}(\Box, \mathit{JUMP})}{\Box q, \Diamond\Box(p \vee \neg p), \Box(p \vee \neg p)}(\Diamond)}{\Box q, \Diamond\Box(p \vee \neg p)}(\vee)}{\Box q \vee \Diamond\Box(p \vee \neg p)}$$

remains an open question

[6]For simplicity, these proofs have been constructed using the naive sequent calculus. The features intended to provide termination for automated reasoning only complicate these fairly simple examples

While the right derivation is basically similar, the left derivation fails, and we must backtrack. Thus, although the first strategy successfully resolves both cases, the second runs into problems in some cases.

However, the cost of marking is the need for more detailed rules, and added complexity in the application of those rules. As the prior method demonstrates, while the distinction between strong and weakly special $\Box$ formulas may have merit, the added complexity of the rules needed almost certainly outweighs the value of the distinction.

Chapter 6

Unavoidable Words

6.1 Introduction

This chapter is not about modal logic. Instead, it discusses the complexity of the unavoidable words problem. This problem is essentially concerned with patterns of repetition in strings of symbols. In particular, the problem, is one of identifying those patterns that must occur in any sufficiently long string.

This problem was first outlined by Thue [41], who applied it to a problem in group theory. He used the concept of avoidablility to outline a sequence. This sequence has been discovered on a couple of other occasions, and tends to show up in a number of different areas, as outlined by Allouche and Shallit [38].

The unavoidable words problem is a subproblem of more general pattern matching problems (see [9] and [21] for examples of more general problems). There is an algorithm suitable for solving the problem, discovered by Bean, Ehrenfeucht and McNulty [3], and independently by Zimin [43]. This algorithm establishes the problem as being firmly in NP. Investigations of this algorithm ([1], [20]) indicate it is unlikely the problem is in P. There are, however, many open problems relating to the unavoidable words [13].

We shall explain the problem, borrowing terms from both [3] and [43] as appropriate. As an altenate reference, the textbook Algebraic Combinatories on Words [28] devotes an entire chapter to this problem. This chapter presents some fairly simple results that follow from the algorithm, as well as an interesting result about the complexity of unavoidability testing for strings that are "long", relative to their alphabet. We then finish by presenting a selection of counter-examples to demonstrate that the problem is indeed hard.

6.1.1 Formal Definitions

We start by defining an **alphabet** A as a set of symbols, and a **word** W on A as a string of symbols taken from A. Note that while a word may be infinitely long, all alphabets are finite sets.

Definition 6.1. The set of all possible words on an alphabet A shall be denoted $\mathfrak{w}(A)$.

Next, we must define what it means to encounter a word:

Definition 6.2. Given a pair of alphabets A and B, a word W on A is a **substitution instance** of word U on B if there exists a function $f : B \to \mathfrak{w}A$, such that if $U = u_1u_2\ldots$ then $W = f(u_1)f(u_2)\ldots$.

Definition 6.3. A word W **encounters** a word U if there is some substring of W that is a substitution instance of U.

If W does not encounter U, we say W **avoids** U.

Because of our use of substitution instances, the exact nature of the symbols constituting an alphabet is largely irrelevant. All alphabets of a given (finite) size may be considered isomorphic. For all our purposes, a word *abcca* would be equivalent to the word *xyzzx*.

We define an unavoidable word as follows:

Definition 6.4. A word U on A is **unavoidable** by an alphabet B if the set $\{W \in \mathfrak{w}B : W \text{ avoids } U\}$ is a finite set.

Note that clearly, if a word W is unavoidable by an alphabet B, we can say that W will also be unavoidable by all alphabets C, where $|C| \leq |B|$. If B has k letters, we say that W is k-**unavoidable**. If W is k-unavoidable for all k, then we call W unavoidable. Words that are not unavoidable are called **avoidable**.

It is not in general the case that k-unavoidability is equivalent to complete unavoidability. For instance, the word xx is unavoidable on a 2 letter alphabet. However, it is possible to avoid xx with any alphabet that has three or more letters [41].

The question of if there exists some number k such that k-unavoidability is equivalent to general unavoidability remains open. Clark [12] demonstrates an example of a word that is 5-unavoidable, which is not unavoidable in general. This is the highest value of k for which k-unavoidability is known not to be equivalent to general unavoidability. It may be possible to derive a larger 6-unavoidable word that is not in general unavoidable, but checking such a word for 6-unavoidability is currently computationally unfeasible.

6.1.2 Decidability

Having defined the property of unavoidability, it is natural to ask if it is in general possible to decide if a word is avoidable. Note that even without a general decidability result, the theory of unavoidable words can still produce useful results. For instance, [41] produces results based on the avoidability of xx, without referencing more complex words.

There are two papers, [43] and [3] that independently established the decidability of the unavoidable words problem. Both use a similar decision procedure, albeit with different notation. The decision procedure is exponential in complexity, but the ultimate complexity of the problem is unknown. Heitsch [22] establishes that most of the 'obvious' candidates for refining the decision procedure are inadequate.

The Zimin Word

One of the important results on unavoidable words is that for any given alphabet W, there are only finitely many unavoidable words that can be created using that alphabet.

In [3], the existence of an upper bound on the length of unavoidable words on an alphabet is established. However, [43] and [37] both go further, establishing that there is a single longest unavoidable word on an alphabet, and it encounters all shorter words on that alphabet. This word is called the Zimin word, and is defined as follows:

Definition 6.5. Let A_n be the alphabet $a_1 \ldots a_n$. Let Z_n be the longest unavoidable word on A_n, as defined below:

$$Z_1 = a_1$$
$$Z_{n+1} = Z_n a_{n+1} Z_n$$

Note that the alphabet A_n is a generalisation for any alphabet on n characters, and substitution instances of Z_n are perfectly acceptable as unavoidable words. For instance, xyx, $a_1a_2a_1$ and bab can all be considered instances of Z_2.

The proof of the unavoidability of Z_n is a simple one. Firstly, it is trivially true that Z_1 is unavoidable. Secondly, any sufficiently long word must encounter an unavoidable word multiple times. Thus, if we define $f(a_{n+1})$ to be the sequence of characters between some two encounters of Z_n, then the word encounters Z_{n+1}. Thus, by induction, Z_n is unavoidable for all n.

Bear in mind that for all Z_i, Z_j, the first m characters of Z_i and Z_j, if defined, will be identical. This fact is used later, in Section 6.3.2.

The Decision Procedure

Both [43] and [3] provide the same basic procedure for determining if a word is unavoidable. This procedure hinges on two basic results. Firstly, we can simplify a word in a way that preserves avoidability - if the original word was avoidable, either the simplification is impossible or the simplified word is also avoidable. Secondly, for any unavoidable word, there is a sequence of simplifications that will eventually reach the word Z_1, which is trivially unavoidable.

To apply the procedure, we start with a set of definitions.

Definition 6.6. R_W is a relationship on the alphabet of the word W. We say aR_Wb iff ab is a subword of W. Thus, R_W defines the relationship of adjacency.

Definition 6.7. Given adjacency, we can create a bipartite graph, where every letter a in the alphabet creates two points a_L and a_R. If we have aR_Wb for some pair a, b, then we create an edge connecting the points a_L and b_R. This is called the adjacency graph of the word. Figure 6.1 provides an example of such a graph.

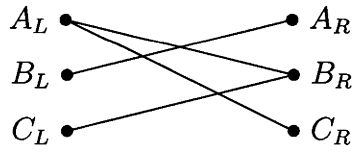

Figure 6.1: Adjacency graph for abacbab

Definition 6.8. A letter a is free in a word W if there is no path in the adjacency graph of W connecting a_L and a_R. For the graph in Figure 6.1, both a and b are free, but c is not.

Definition 6.9 (Deletion)**.** Given a word W and a letter a, let $W - a$ be the word created by removing all instances of the letter a from W. For instance, if $W = xyxzyxy$, $W - y = xxzx$.

Definition 6.10 (Unification)**.** Given a word W and a set of letters σ, let $W\sigma_a$ be the word created by replacing every occurrence of a letter in σ by the letter a. For instance, if $W = xyxzyxy$, $\sigma = \{x, z\}$, $W\sigma_a = ayaayay$.

Definition 6.11. A word is locked if it contains no free letters. For example, the word *cabadcb* has the adjacency graph shown in Figure 6.2, and is locked.

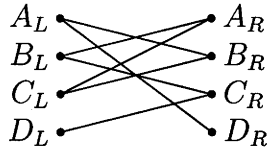

Figure 6.2: Adjacency graph for the locked word cabadcbc

Now, both [43] and [3] show that given an unavoidable word, there exists a sequence of unification and deletion of free letters that will reduce the word to Z_1 (and the only possible operation on Z_1 is a deletion to reduce it to the empty string). If a word is avoidable, then any sequence of unification and deletion operations must eventually produce a locked word.

Since both unification and deletion reduce the number of different letters in the word and there are only finitely many options at each stage, there are only a finite number of possible sequences to consider. It is simple to see that the process must always terminate, and so the unavoidable words problem is decidable.

6.2 Other definitions

We can simplify the working for our problem by defining a number of other operations and objects.

It is a feature of the algorithm that letters need only be unified immediately before deletion. This can be seen from the definition of free letters - unification cannot *make* a set of letters free. Thus, postponing unification until it is necessary to delete multiple letters at once cannot prevent us from discovering any necessary reductions. Because of this, it is convenient to replace our two operations of unification and deletion with a single operation, reduction by free sets.

Definition 6.12. A free set is a set of letters σ used in a word W, with the property that a is a free letter in the word $W\sigma_a$.

Definition 6.13. Given a word W and a set of letters σ, $W-\sigma$ is the word created by deleting all instances of all letters in σ. We call this operation reduction by the free set σ.

A point to note here, originally made by Heitsch [21] is that if a word is unavoidable, then there exists a reduction by free sets that uses only free sets with the property that $\forall x, y \in \sigma$, there is a chain of letters $z_1 \dots z_n$, with $xRz_1R^{-1}z_2 \dots Rz_nR^{-1}y$.

Translated into the terms of unification/deletion relations, this result means that unification is only necessary if performing the unification does not change the collection of free sets in the word.

6.3 Basic Results

The following results are all fairly simple consequences of the correctness of the decision procedure. However, they are often not mentioned in the literature, and can be useful when trying to work with unavoidable words.

6.3.1 Singletons

A singleton is a letter that occurs only once in a word. These letters have a number of special properties. Individually, these properties make it simpler to work with unavoidable words under some circumstances. Taken as a whole, they lead to the equivalence between unavoidable words and unavoidable formulas cited in [12].

Firstly, there is the trivial note that as with any other letter, the identity of singletons doesn't matter. The words *abacaba* and *abadaba* are essentially the same word, when checking for unavoidability. Now, when it is necessary to check the interaction of words, the identity of singletons becomes important - *aba* and *bab* behave in identical fashion in isolation, but while *abacaba* and *babcbab* are both unavoidable, the word *abacbab* is avoidable.

Secondly, we augment this result with the following theorem:

Theorem 6.14. *A word W can be reduced to an empty string by a sequence of reductions if and only if it can be reduced to a word S which contains every singleton in W, and only singletons.*

Proof. To prove this, we rely on the reduction procedure, and note that it is never a disadvantage to leave a singleton out of a reduction; If σ_1 is a free set in $W - (\sigma_2 \cup \{s\})$, where s is a singleton, then σ_1 is a free set in $W - \sigma_2$. Thus, if we take a valid reduction of W to an empty word, removing any singletons from

the reduction sets produces a reduction of W to S. From here, a simple inductive argument suffices to prove the hypothesis.

That singleton reductions are never necessary for a set to become free follows from the definition - if removing the letter w from a word would make a set σ_1 free, then there is some pair x, y in the adjacency graph such that $x_r R w_l R^{-1} y_r$. But this is a statement that the word contains the pairs xw and yw, which would require w to occur multiple times.

The converse is trivial - any word S containing only singletons can be reduced to the empty string by a few extra reductions. ⊣

Now, given these two results, we can justify abbreviating singletons, using only a single character '.' that is used for every singleton in the word; *abxacybc* becomes *ab.ac.bc*. We do not need to include the . character when building adjacency graphs, and when the word has been reduced to ..., we can conclude unavoidability.

We next note results that are in a way opposite to the prior result. Having noted that removing singletons is never necessary, we now note that there are some circumstances when singletons can be removed immediately, to simplify a word.

Firstly, observe that when singletons occur together, as in the word *abaxyaba*, we can remove one singleton immediately, producing *abaxaba* (or, equivalently, *abayaba*) without affecting the unavoidability of the word.

Further, singletons at the beginning and end of words can be removed immediately. In a word like *xabacabay*, the x and y play no part in the unavoidability of the word.

These results all hinge on the nature of reductions - a word has no valid reductions only if all suitable free sets have been locked. As outlined previously, a set σ is locked iff it has a path within the adjacency graph connecting x_L to y_R for some $x, y \in \sigma$. In the adjacency graph, singletons must be the endpoints of a path, and so irrelevant for any set unless they are a member, and by our previous theorem, it is possible to remove the singleton from any necessary free set. Thus we justify considering the word *xabaycabzbad* as being identical to *aba.ab.ba*, which has a much clearer structure.

6.3.2 The Zimin word

The Zimin word is an interesting one, and there are some basic results connected to it. Having these results can make it easier to work with other unavoidable

words, as Z_n is highly structured, and all unavoidable words have some connection to Z_n.

The Zimin sequence

While the Zimin word is defined recursively, it is good to have an iterative definition, such that we can tell (for example), the 35th letter of Z_6, without running through the other letters of the 64 character word. Thus, we define a sequence z_n, called the Zimin sequence, over the (infinite) alphabet $a_n, n \in \mathbb{N}$:

Definition 6.15. $z_k = a_{f(k)}$, where:
$f(2i-1) = 1$
$f(2i) = f(i) + 1.$

The letter z_n corresponds to the n-th letter of the Zimin word. Thanks to its recursive definition, the prefix of the Zimin word Z_n is identical for all n.

For most practical values of z_n, we can map the sequence from the natural numbers to a more normal alphabet $a, b, c, \ldots$ without running out of letters. Words on more than 10 letters are difficult to handle, and there is a combinatorial explosion as more letters are added. Running out of letters when trying to write 32 million character words is not going to come up within this thesis.

Subsequences of the Zimin word

All unavoidable words on n characters are subsequences of Z_n. (Not necessarily a unique subsequence.) For example, the word *abcab* can be mapped to a subsequence of Z_3 in two ways, after renaming:

$Z_3 =$ *xyxzxyx*
ab cab
abc ab

Interestingly, this is never mentioned in Heitsch [22], although the proofs provided imply this fact, indirectly.

Indeed, using the splitting algorithm provided in Heitsch [22], it is possible to get the relevant subsequence of Z_n in trivial fashion. The splitting provides an ordering on letters, and this ordering corresponds to the ordering based on the frequency of letters in Z_n.

This makes enumerating the possibilities for extremely long words relatively simple, allowing us to extend the results of Schmidt [37], establishing an upper bound on the number of words of length $2^n - m$ for any $m \leq 2^{-1}n$. Further, this

upper bound is constant as n changes, and for sufficiently large n, is a complete enumeration.

To do this, we simply exploit the need for the Zimin word to encounter this subsequence. We can, for instance, create all words on $2^n - 2$ characters by taking the Zimin word, and removing a character. Since the new word must encounter the Zimin word, every instance of a given letter must have the same set of deletions surrounding it. When deleting only a single character, we can either delete from the ends of the word, or from the characters surrounding the singleton in the middle. The singleton in the middle is the only character that can be expanded by a homomorphism to cover the deletion without also expanding a character without an associated deletion

For instance, if we take $Z_4 = abacabadabacaba$, and delete the third character, to produce *abcabadabacaba*, then the only homomorphisms that would take the left hand side of the word to *abacaba* are $f(b) = ba$ and $g(c) = ac$ (Both f and g act as an identity mapping on the other characters in the word). But under f, we get the word *abacabaadabaacabaa*, which is clearly avoidable, while under g, we get *abacabadabaacaba*, again, including the avoidable word *aa*.

So, with this understanding, we can consider the options for deleting two characters from Z_4, as shown in the following list, with the characters to be deleted shown in bold:

*abacabadabaca***ba**
ab*acabadabacaba*
a*bacabadabacab***a**
*abaca***ba***dabacaba*
*abacabad***ab***acaba*
*abacab***ad***abacaba*
a*bacab***ad***abacaba*
a*bacabad***ab***acaba*
*abacab***ad***abacab***a**
*abacabad***ab***acab***a**
*abac***a***badabac***a***ba*
*ab***a***cabadaba***a***caba*

After performing the marked deletions, and removing those patterns that are equivalent under reversal, we get the 7 unavoidable patterns in Schmidt.

For further enumeration of the upper bounds on the number of unavoidable words of length $2^n - m$, see Appendix A, which shows the result of some simple

brute force calculation for values of m up to 25.

This is an upper bound, rather than an exact enumeration, because, for small n, deletion from one point may be indistinguishable from deletion from another point. For instance, deleting 2 characters from *abacaba*, if we remove both letters to the right of a *b*, we shall reduce the word **a***bac***a***ba* to *bacba*. If we instead a letter to the left of *c* and the final character from the word, we reduce *abac***a***b***a** to *abcab*, an identical pattern. However, for sufficiently large n, there will always be the full set of words.

Splitting words

As previously mentioned, [22] offers a polynomial time algorithm that accepts all unavoidable words and many avoidable ones, and provides a "split-based" ordering on letters. The algorithm, applied to a non-empty word W on n letters, is as follows:

Definition 6.16. Begin with $X = \{W\}$.
If $\forall U \in X$, U is empty, accept the word.
Else if there exists l such that for all $U \in X$, either l is not in U, or $U = VlV'$, where l is not in V or V', then repeat with the set $X' = \{U|U \in X, l \notin U\} \cup \{V|VlV' \in X\} \cup \{V'|VlV' \in X\}$.

For a proof that this accepts all unavoidable words, but does not accept only unavoidable words, see [22]. However, if we keep track of our choice of l at each step, then we produce an ordering on letters. Most usefully, if the choice of l was unique at each step, then the reverse of this ordering is the only possible valid reduction on W. This single reduction is quickly checked, and goes some small way to reducing the number of false positives encountered by the algorithm.

6.4 Simplifications

The unavoidable words problem is a difficult one. In this section, we look at some simpler, yet closely related problems.

6.4.1 Simply reducible words

A simple reduction is one that involves no unification steps - The word can be reduced entirely by reducing single characters. Such words are far easier to work

with, and avoid one of the major causes of combinatorial explosion when checking for unavoidability.

6.4.2 Long unavoidable words

Many of the hardest words to show unavoidability are the ones that are "short", relative to the number of characters in the word. A word with many short blocks, separated by singletons, has more options at each stage of the reduction. By contrast, if looking for valid reductions for Z_n, then there is only one valid sequence of reductions, and attempting to remove a letter out of order immediately results in a trivially unavoidable word.

The longer a word gets, the closer it grows to the Zimin word, and the easier it must be to reduce.

An equivalence result

Theorem 6.17. *All sufficiently long words are simple words, in the sense of the previous section. For this purpose, "sufficiently long" means words on n letters of length at least* 2^{n-1}.

Proof. We prove this by induction.

The base case of the induction, words on only one or two letters, is essentially trivial.

For the inductive step, assume that for all $m < n$, words on m characters of at least length 2^{m-1} are simply unavoidable. Take an unavoidable word on n letters of length at least 2^{n-1}. Since it is unavoidable, there exists either a unification or reduction operation that will take it to a simpler unavoidable word.

If we apply a unification operation to the word, we have a word on at most $n-1$ letters that is longer than the Zimin word, and is as such trivially avoidable.

Therefore, there must be a single letter reduction to take it to a simpler unavoidable word. This single letter must be free, which means that it can occur at most every second character (otherwise, we have a subword of the form aa)

Because of this, if our original word has length $l \geq 2^{n-1}$, then the resulting word after a single letter reduction will have length at least $\lfloor l/2 \rfloor \geq 2^{n-2}$, which is a long word on $n-1$ characters, and so we can apply the inductive hypothesis. ⊣

6.5 Implied Unavoidability

Given that a word W is unavoidable, it is often possible to immediately determine that some other words are also unavoidable. As a simple example, all subwords of W will be unavoidable. This kind of implication can be useful when trying to create lists of unavoidable words, as it allows for various *a priori* methods to be applied:

- The most basic implication has already been discussed - We can turn all instances of one letter into a different letter, not occurring in the word, without affecting unavoidability. Thus, aba is an unavoidable word iff xbx is an unavoidable word.

- By the same token, the discussion of singleton pruning leads to the realisation that introducing new singletons to a word cannot make the word avoidable. However, they can make an avoidable word unavoidable. For an extremely simple example, aa is avoidable, but adding b can produce the unavoidable aba.

- A necessary consequence of the algorithm is that a word $w_1 \dots w_n$ being unavoidable is equivalent to the reversed word $w_n \dots w_1$ being unavoidable, with the same valid reductions. At each stage of the reduction, the adjacency graph of one will be the mirror image of the adjacency graph of the other, and so they will permit the same set of free sets.

- Also, there is a trivial result that states all subwords of an unavoidable word are unavoidable. This result is quite useful in the inverse form, stating that to be an unavoidable word, it is necessary that all subwords must be unavoidable. Because of this, locating an avoidable subword suffices to show that an entire word must be avoidable.

  Because of this result, the most "difficult" avoidable words to detect are of the form $W = xUy$, where both xU and Uy are unavoidable. If there were some easy way to detect unavoidable words of this form, then the entire problem would be simpler.

- It was shown by Zimin [43] that all unavoidable words on n characters encountered Z_n, and, as with subwords, this can theoretically be used to identify avoidability, by finding a homomorphism to map a word W to Z_n.

Indeed, the unavoidable words problem can be more simply viewed as a problem of identifying those words the Zimin word encounters.

However, the problem of encountering in general is itself a very difficult problem. Because of this, it may well be the case that identifying words encountered by the Zimin word is also an NP-complete problem - its complexity is currently unknown.

- The operation of unifying two distinct letters does not introduce unavoidability. The contrapositive of this is that taking a letter and replacing some (but not all) instances of the letter with a new letter does not introduce avoidability, since the change cannot reduce the available free sets and the resulting word can simply unify the two letters when the need arises.

- The operations involved in reduction are not affected by (suitably separated) duplication of a subword. For instance, if we have a reduction for the word *aba*, then this reduction is a valid reduction to reduce the word *abacaba* to *c*.

  As such, if we have an unavoidable word X, then $X.X$, where . is a singleton not occurring in X, is also an unavoidable word.

- To generalise the previous idea, given two unavoidable words X and Y, if Y is either a subword of X or the overlap between the alphabet of X and the alphabet of Y is empty, then $X.Y$ is an unavoidable word.

- The general replacement operation introduced by Heitsch [22], preserves unavoidability. This is an especially potent operation, since it makes it relatively easy to generalise examples. Given a single counterexample to a conjecture, this operation can be used to generate an infinite number of distinct counterexamples.

- A final operation that will create a new unavoidable word from an old unavoidable word is of course adding a free set, since that operation is trivially reverseable when looking for a reduction.

6.6 Counterexamples

There are a number of conjectures about the unavoidable words that are not obviously false. However, if they were true, they could make the unavoidable

words problem much easier. Presented here are a number of these conjectures, and the counterexamples we found:

6.6.1 Unification as the first step

The algorithm, as presented, involves two kinds of operation, unification and reduction. It is immediately clear that for some words, it is not necessary to reduce immediately after unifying. One may conjecture that given a sequence of reduction and unification operations, it is possible to rearrange these operations such that the unification operations are complete before the first reduction operation. If true, this could make it possible to turn every unavoidable word into a simple word by a process of repeated unification.

The counterexample to this is the word *xaxbxaxcdebecb* The only valid reduction for this is to reduce by x, then unify a and e. However, if we unify a and e before reducing by x, then x is no longer a free set, and the resulting word *xaxbxaxcdxbxcb* is avoidable.

6.6.2 Partial reduction

Given an unavoidable word X, and a free set σ, such that $X - \sigma$ is unavoidable, it is tempting to make conjectures about the word $X.(X - \sigma)$. Unfortunately, there is no clear result.

It is not invariably avoidable, as is made clear by such trivial examples as *aba.b* and *xabax.aba*. However, it is not invariably unavoidable. An example of an avoidable word of this form is provided by *abacaba.bcb*.

6.6.3 Free-set size

The unavoidable words problem would be made far easier if there were some easy way to determine the appropriate free set. One tempting strategy to take is to look at free set size. Alas, this is inadequate. There are words where the smallest free set is not the appropriate next reduction, such as *abacdebecb*, and there are words where the largest free set is not the appropriate next reduction, such as *abacabadxax*.

For a more detailed explanation of this failure, see Heitsch [21].

6.6.4 Simple strings

Call a simple string one in which no character is repeated. It is clear that a word can be made entirely of simple strings, separated by singletons. *abc.acb* is one example of an unavoidable word with such a property, and *ab.ba.bc.cb.ca* is an example of a locked word with such a property. While such words seem simpler than the general problem of unavoidable words, it is not in general the case that such words must be locked or unavoidable. *axb.bax.bc.cb.cax* is an example of a word that is avoidable, in spite of having a reduction. Both a and x are free letters, however, it clearly encounters the locked word *ab.ba.bc.cb.ca*.

6.6.5 Splitting entails a reduction

As previously mentioned, when using the splitting algorithm defined in Heitsch [22], if the resulting ordering is unique, and the word is unavoidable, then the ordering is identical to that provided by the reduction. However, it is simple to verify that when a unification step is necessary in the reduction of a word, the ordering must not be unique.

Given that the long unavoidable words never require a unification step in their reduction, we could conjecture that the ordering supplied will always correspond to a valid reduction in this particular case. That is to say, the only reason that the algorithm does not produce a unique splitting is because of the inability to impose an order on letters that must be unified. The counterexample here is the word *yaxbyxax*, which has multiple distinct orderings based on the splitting, and is a long word, where any unification will produce a word too long to be unavoidable. But, the word only has one valid reduction, namely y, x, a, b.

6.6.6 Ordering reductions

The standard way of demonstrating that a problem is NP-complete is to embed some other problem, known to be NP-complete, into the problem. This is not easy. For the unavoidable words problem, there would need to be some way of limiting valid reductions that didn't trivially produce over constrained reductions.

One idea is to attempt to impose order on the reductions. It is trivial to see that if *aba* is a subword of an unavoidable word, then in any valid reduction, a must be reduced before b, lest the reduction created the avoidable subword *aa*.

While ordering in this manner is possible, it is immediately obvious that overconstraining is an issue; *aba.bcb.aca* is a locked word. If one were to place

an ordering on multiple letters (that is, a before b before c), then adding the transitive closure of this ordering can trivially lock the word.

However, we can reduce the amount of detail used to create the ordering. This delays, but does not fix the problem. Take the basic partial order $A > B > C > D$ and $A > E > D$. Attempting to impose this on a reduction by creating the word *aba.bcb.cdc.aea.ded* produces a locked word. This is a common issue when attempting to embed problems into the unavoidable words problem; Even though a problem might embed easily for small cases, the non-local nature of locking can produce a locked word when extended to a slightly more complex problem.

On the other hand, creating orderings using the Zimin word has its own problems. While *abacabadabacaba.aeadaea* is not locked, and is indeed unavoidable, the immediate issue is that the Zimin word grows exponentially in length as additional constraints are added. It can hardly be considered an "easy" way of ordering reductions.

Chapter 7

Conclusions and Further Work

This thesis has covered four topics, with an emphasis on modal logic. In this chapter, we shall review the results provided in earlier chapters and suggest directions for further work.

Chapter 3 discusses the structure of the lattice of normal extensions of **KTB**. Thanks to the duality of logic and algebra, this is also the lattice of subvarieties of KTB. The lattice has a few known structural properties. Of particular interest to us, the lattice of normal extensions has a unique greatest element, the logic of a single reflexive point, and a unique second greatest element, the logic of two connected reflexive points.

Our work has two theorems discussing the set of third greatest elements in this lattice. The first, simpler theorem is Theorem 3.7. This theorem demonstrates a construction of infinitely many cocovers of the algebra of two points. While this theorem is unnecessary in light of later results, it is significantly simpler to prove than subsequent theorems, relies only upon Kripke frames, while the subsequent theorem requires the use of general frames. Thus, the result retains some theoretical interest.

The second major theorem of Chapter 3 is Theorem 3.17. This is a more complex theorem than the first, and requires substantially more effort to prove. However, it is still a constructive proof, this time that there are uncountably many cocovers of the algebra of two points. Thus, this theorem is strictly stronger than Theorem 3.7.

The primary reason that the second theorem is so much more complex than the first is that to demonstrate the existence of uncountably many cocovers, we must work with infinite structures. At this point, we are forced to use general frames instead of Kripke frames. For a general frame, having added a set of distinguished

points to a Kripke frame, we must deal with the possibility of a logic based upon the same Kripke frame, with a different set of distinguished worlds. Fortunately, by placing a bound on the diameter of the frame, we successfully limited the potential complexities involved.

In the algebra dual to our infinite frame, this bound on the diameter of the frame gives us a simple way to deal with ultraproducts when discussing the variety that the algebra generates. This makes the proof substantially simpler. Indeed, the bulk of the proof is a relatively simple argument of cases, since with ultrapowers omitted, we need only worry about direct subalgebras, and these can be divided into a few well-defined cases.

Since **KTB** only has uncountably many normal extensions, we may conclude that the second theorem is as strong as possible. There is no room for a theorem that shows even more cocovers of the algebra of two points.

While the result for **KTB** may not be improved, further work could look at other, related logics to see if a similar result could be obtained. One suggestion is the set of axioms alt_n. These limit the potential branching of our frames. Since the results obtained rely on having a point that can see arbitrarily many other points, the question for extensions of $\mathbf{KTB} \bigoplus alt_n$ remains unanswered.

However, such a proof would be substantially more complex. The proof for **KTB** simplified the problem by placing a bound on the diameter of the infinite frames we dealt with. However, a frame with finite diameter and finite branching is itself finite. For $\mathbf{KTB} \bigoplus alt_n$, if it were possible to show uncountably many cocovers of the algebra of two points, then we would need to cover the potential disruption caused by ultrapowers.

Normal forms are a tool for showing that a logic has the finite model property or Kirpke completeness. Chapter 4 was inspired by noticing a mistake in [32], regarding normal forms for **KTB**. The chapter was written to fix that error, and explain some pitfalls that could lead to such an error. This chapter presents normal forms for directed frames, connected frames, symmetric frames and frames with restricted branching. It details the process of creating models from normal forms.

Most importantly, the chapter looks at how the various normal forms combine when multiple axioms are added to a logic, using the example of $\mathbf{KTB} \bigoplus alt_n$ to show that while the modifications can often be done, they are not always trivial. The further example of n-transitivity shows an example of a case where normal forms are not applicable.

While it is quite possible to spend time producing appropriate normal forms

for any well-behaved logic, this is not an area of great interest for further research. Far more interesting is the pursuit of general results such as those provided in [16], [34] and [18].

The greatest strength of normal forms is their strong association between worlds in a model and particular formulas. This makes reasoning from the structure of a formula to the structure of a model significantly easier, and enables the creation of useful theoretical results. However, normal forms are not a practical tool of reasoning, as a sequent calculus can be. The structure of a normal form is prone to exponential growth, and trying to represent a particular formula in normal form is not generally practical. It is worth noting that most works on normal forms are content with existence proofs, and avoid trying to write out normal forms in full.

Chapter 5 compared two ways of creating a sequent calculus for the logic **S4** that terminates when used in a "backward" manner. On the one hand the established method of histories works by keeping multiple formula sets. These extra formula sets are used to carry along a record of prior rule applications, and terminate a derivation before it enters an infinite loop.

The alternative is the new concept of mark and index based methods, claimed to be sound and complete in [36]. These add markup to the formula set to try and prevent the non-termination issues that arise in a regular sequent calculus. However, as presented, the marks added lack the context needed to guarantee both termination and completeness, which indicates significant flaws in the claimed proofs of [36]. While we have shown it is possible to create a mark and index based calculus that is essentially identical to the history based method, we have failed to find any compelling situation where the marks and index based method offers an actual improvement when compared to history-based methods, and recommend not using mark and index based methods in future.

The unavoidable words problem is a very difficult problem. A significant portion of Chapter 6 is dedicated to showing techniques that do not work when applied to the problem.

A major positive result of the chapter is Section 6.3.2 an extension of [37], explaining the structure of long unavoidable patterns. With this, we provide a complete description of the upper end of the problem, and it is only the (more difficult) lower end that remains open.

Further, Theorem 6.17 explains exactly how the long unavoidable words are simpler than short words. The unavoidable words problem can be resolved by the operation of deletion by free sets. For long words, the necessary free sets are

strictly single element sets, while short words can be created that need arbitrarily large free sets to successfully reduce.

It should be possible to extend the result enumerating the long unavoidable patterns to also provide some enumeration of the shorter unavoidable words. This would provide a relatively efficient way of generating all the unavoidable words, and while still too complex to be a practical decision procedure, it would give a much clearer description of the growth of the problem.

The best avenue for further research in the field of unavoidable words probably lies in k-unavoidability. There are suggestions of interesting patterns in [12], and computers have improved noticeably over the years, so the search for a 6-unavoidable word may be computationally feasible, especially if there is some useful simplification/pattern to be found.

Appendix A

Enumeration of long words

This table shows the upper bound on the number of unavoidable words of length 2^n-m, for m up to 25. Note that for compatibility with the results of Schmidt [37], symmetric words are considered identical; both *abacab* and *bacaba* are considered the same word. Actually attaining the upper bound may require an extremely large alphabet, so higher values of m lack some practical interest.

m	Maximum number of unavoidable words with length $2^n - m$
1	1
2	2
3	7
4	14
5	32
6	58
7	109
8	182
9	307
10	482
11	757
12	1134
13	1692
14	2442
15	3503
16	4902
17	6816
18	9298
19	12605
20	16830
21	22340
22	29290
23	38191
24	49286
25	63281

Appendix B

Some Finite Graphs Covering $V(\mathfrak{K}_2)$

To create these finite graphs, a simple C program was written to check for subgraphs, and this was connected to Brendan McKay's nauty program [30] to generate sets of graphs of an appropriate size.

After this, creating the images was a matter of patience and a decent graphics program.

First, we show all the graphs covering $V(\mathfrak{K}_2)$ with 8 points:

Secondly, we show a small subset of the 10 and 11 point graphs - The number of graphs covering $V(\mathfrak{K}_2)$ on a given number of points grows rapidly, from the 8 graphs on 8 points to 849 graphs on 10 points, and far too many to calculate easily on 12 points.

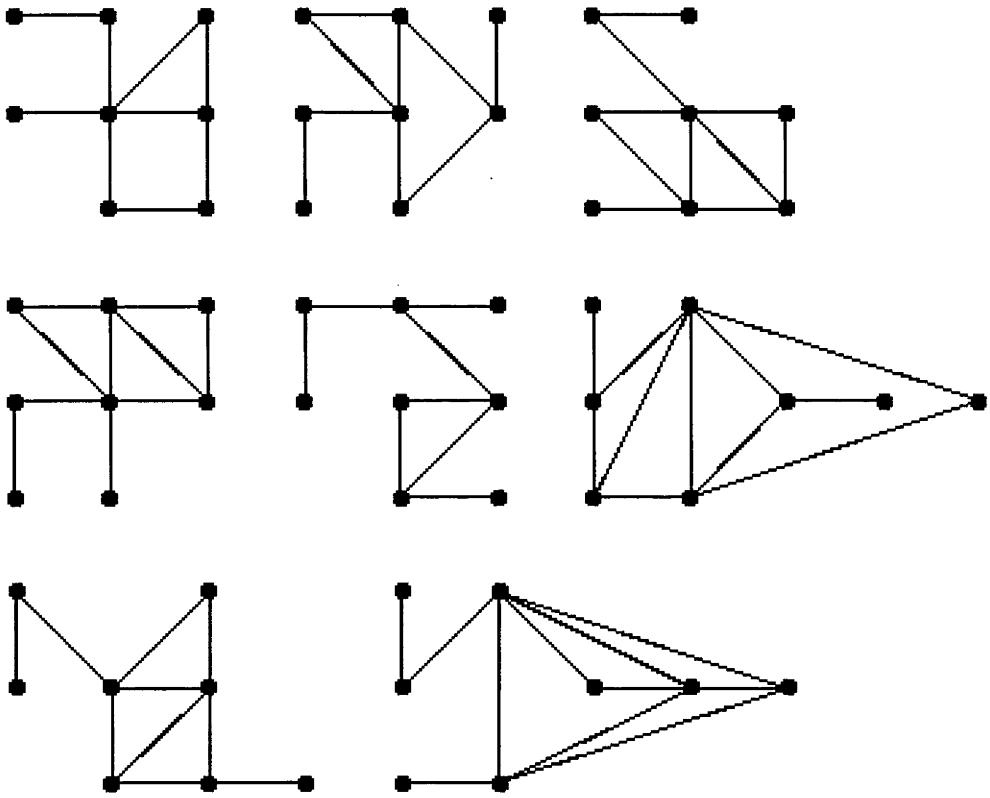

Figure B.1: All 8 point covers of $V(\mathfrak{K}_2)$

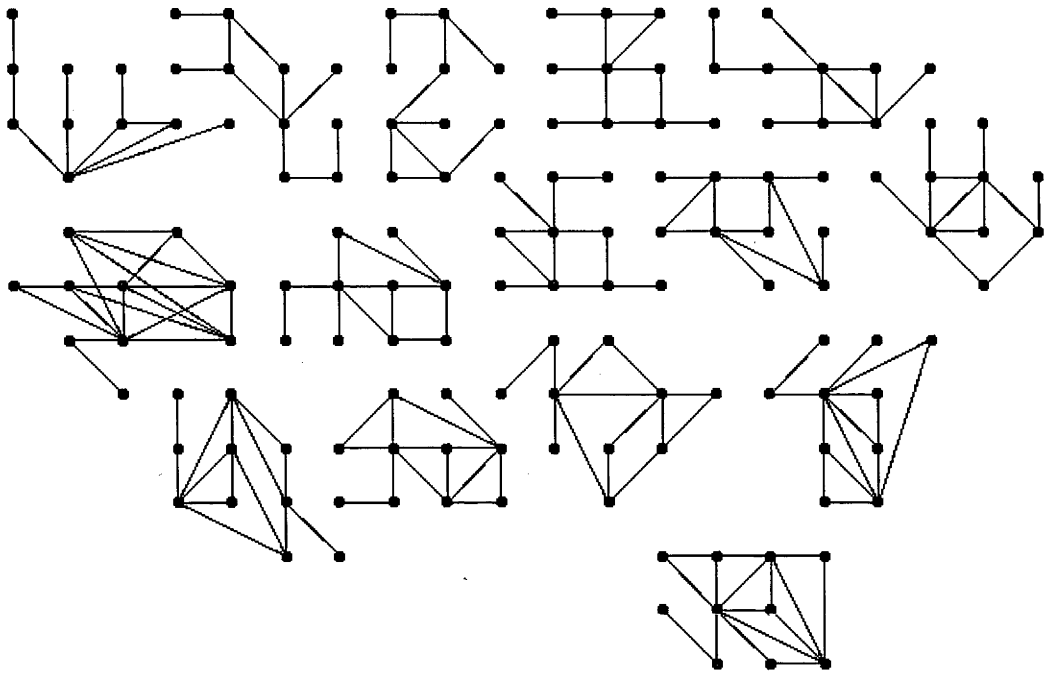

Figure B.2: Some covers of $V(\mathfrak{K}_2)$ on 10 points

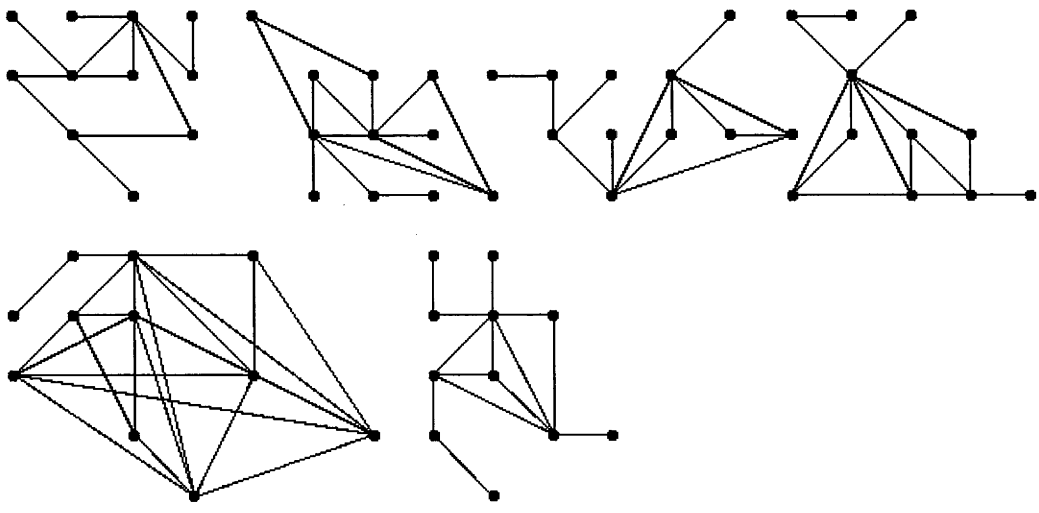

Figure B.3: Some covers of $V(\mathfrak{K}_2)$ on 11 points

Bibliography

[1] K. A. Baker, G. F. McNulty, and W. Taylor. Growth problems for avoidable words. *Theoretical Computer Science*, 69:319–345, 1989.

[2] P. Balsiger, A. Heuerding, and S. Schwendimann. A benchmark method for the propositional modal logics K, KT, S4. *Journal of Automated Reasoning*, 24:297–317, 2000.

[3] D. R. Bean, A. Ehrenfeucht, and G. F. McNulty. Avoidable patterns in strings of symbols. *Pacific Journal of Mathematics*, 85(2):261–294, 1979.

[4] B. Bennett. Modal logics for qualitative spatial reasoning. *Bulletin of the Interest Group in Pure and Applied Logic (IGPL)*, 4(1):23–45, 1996. WWW address ftp://ftp.mpi-sb.mpg.de/pub/igpl/Journal/V4-1/index.html.

[5] G. Birkhoff. Subdirect unions in universal algebra. *Bull. Amer. Math. Soc.*, 50(10):764–768, 1944.

[6] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge tracts in theoretical computer science.* Cambridge University Press, 2002.

[7] P. Blackburn, J. van Benthem, and F. Wolter, editors. *Handbook of Modal Logic.* Elsevier, 2007.

[8] S. Burris and H. Sankappaanavar. *A Course In Universal Algebra.* Number 78 in Graduate Texts in Mathematics. Springer-Verlag, 1981.

[9] A. Burstein and S. Kitaev. On unavoidable sets of word patterns. *SIAM Journal on Discrete Mathematics*, 19(2):371–398, 2005.

[10] A. Chagrov and M. Zakharyaschev. *Modal Logic.* Oxford Science Publications, 1997.

[11] C. L. Chen, P. S. Grisham, S. Khurshid, and D. E. Perry. Design and validation of a general security model with the Alloy Analyzer. 2006.

[12] R. J. Clark. *Avoidable Formulas in Combinatorics on Words.* PhD thesis, University of California, Los Angeles, 2001.

[13] J. Currie. Open problems in pattern avoidance. *The American Mathematical Monthly*, 100(8):790–793, October 1993.

[14] H. B. Curry. The elimination theorem when modality is present. *The Journal of Symbolic Logic*, 17(4):249–265, December 1952.

[15] K. Fine. An incomplete logic containing S4. *Theoria*, 40:23–29, 1974.

[16] K. Fine. Normal forms in modal logic. *Notre Dame Journal of Formal Logic*, 16(2):229–237, April 1975.

[17] D. Gabelaia, A. Kurucz, and M. Zakharyaschev. Products of transitive modal logics without the (abstract) finite model property. In *Proceedings of AiML 2004*, September 2004.

[18] S. Ghilardi. An algebraic theory of normal forms. *Annals of Pure and Applied Logic*, 71:189–245, 1995.

[19] R. Goré. *Cut-free Sequent and Tableau Systems for Propositional Normal Modal Logics.* PhD thesis, University of Cambridge, November 1991.

[20] C. E. Heitsch. Exact distribution of deletion sizes for unavoidable strings. In *Proceedings of the 8th International Symposium on String Processing and Information Retrieval*, Laguna de San Rafael, Chile, November 2001. IEEE Computer Society Press.

[21] C. E. Heitsch. Generalized pattern matching and the complexity of unavoidability testing. In *CPM '01: Proceedings of the 12th Annual Symposium on Combinatorial Pattern Matching*, pages 219–230, London, UK, 2001. Springer-Verlag.

[22] C. E. Heitsch. Insufficiency of four known necessary conditions on string unavoidability. *Journal of Algorithms*, 56(2):96–123, 2005.

[23] A. Heuerding. *Sequent Calculi for Proof Search in Some Modal Logics.* PhD thesis, Universität Bern, 1998.

[24] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Theorem Proving with Analytic Tableaux and Related Methods*, volume 1071 of *LNCS*, pages 201–225. Springer Berlin / Heidelberg, 1996.

[25] B. Jónsson. Algebras whose congrence lattices are distributive. *Mathematica Scandinavica*, 21:110–121, 1967.

[26] T. Kowalski and Y. Miyazaki. All splitting logics in the latice NExt(KTB). *Trends in Logic*, 27:1–15, 2008.

[27] D. Kozen and R. Parikh. An elementary proof of the completeness of PDL. *Theoretical Computer Science*, pages 113–118, 1981.

[28] M. Lothaire. *Algebraic Combinatorics on Words*, volume 90 of *Encyclobedia of Mathematics and its Applications*. Cambridge University Press, 2002.

[29] D. Makinson. Some embedding theorems for modal logic. *Notre Dame Journal of Formal Logic*, 12(2):252–254, April 1971.

[30] B. McKay. nauty. http://cs.anu.edu.au/ bdm/nauty/. a program for computing automorphism groups of graphs and digraphs.

[31] Y. Miyazaki. The structure of the lattice NExt(KTB). Trends in Logic III International Conference in memoriam Andrzej Mostowski, Helena Rasiowa, Cecylia Rauszer, September 2005.

[32] Y. Miyazaki. Normal forms for modal logics KB and KTB. *Bulletin of the Section of Logic*, 36:3/4:183–194, 2007.

[33] J. D. Monk and R. Bonnet, editors. *Handbook of Boolean Algebras*. Elsevier, 1989.

[34] L. S. Moss. Finite models constructed from canonical formulas. *Journal of Philosophical Logic*, 36(6):605–640, December 2007.

[35] V. Padmanabhan and G. Governatori. A fibred tableau calculus for modal logics of agents. In M. Baldoni and U. Endriss, editors, *DALT*, volume 4327 of *Lecture Notes in Computer Science*, pages 105–122. Springer, 2006.

[36] R. Pliuškevičius and A. Pliuškevičienė. A new method to obtain termination in backward proof search for modal logic S4. *Journal of Logic and Computation Advance Access*, November 2008.

[37] U. Schmidt. Long unavoidable patterns. *Acta Informatica*, 24:433–445, 1987.

[38] J. O. Shallit and J.-P. Allouche. The ubiquitous prouhet-thue-morse sequence. In C. Ding, T. Helleseth, and H. Niederreiter, editors, *Sequences and Their Applications: Proceedings of SETA '98*, pages 1–16. Springer-Verlag, 1999.

[39] M. H. Stone. The theory of representation for boolean algebras. *Transactions of the American Mathematical Society*, 40(1):37–111, July 1936.

[40] A. Tarski. A remark on functionally free algebras. *Annnals of Mathematics*, 47(1):163–166, 1946. January.

[41] A. Thue. Über unedliche zeichenreihen. *Norske Vid. Selsk. Skr., I. Mat. Nat. Kl., Christiana*, 7:1–22, 1906.

[42] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Number 43 in Cambridge Tracts in Theoretical computer science. Cambridge University Press, 2nd edition, 2000.

[43] A. Zimin. Blocking sets of terms. *Math. USSR Sbornik*, 47(2):353–364, 1984.

Index