MODAL LOGICS, CORRESPONDENCE THEORY AND SECOND-ORDER QUANTIFIER ELIMINATION METHODS

A dissertation submitted to the University of Manchester
for the degree of Master of Science
in the Faculty of Engineering and Physical Sciences

2011

By
Christopher S Hampson
Mathematics

Contents

Word Count: 19,054

List of Tables

List of Figures

Abstract

This dissertation focuses on a variety of modal logics, an extension of propositional logic by 'modalities' which are not truth functional. The basic modal logic, containing a single modality, will be our main focus, but we will also discuss various extensions, such as the difference operator, global modalities, graded modalities and hybrid modal logic. Due to their semantic interpretation in so-called Kripke frames, modal logics have a natural translation into second-order logic but, what is more interesting, is the connection that modal axioms have with first-order properties. We can, for example, express the properties of reflexivity, transitivity and euclideanness with relatively simple modal axioms. However, other modal axioms, such as McKinsey's axiom, do not correspond to any first-order property at all.

We discuss several procedures for computing first-order correspondence properties for a large class of modal axioms of the basic modal logic and the relationships between them. We introduce extensions to these procedures which increase their computational capabilities to an even larger class of modal axioms, whose correspondence properties lie in the first-order logic of fixed-points. We remark on the applicability of two of these procedures to languages other than the basic modal language and develop an extension of the third, in order to be applicable to languages involving the difference operator.

Lastly, we explore the modal logics with second-order propositional quantifiers - an area of research that remains largely untouched apart from a few sparse papers in the literature. We describe a class of formulas with first-order correspondence properties that can be effectively computed. Finally, we show, via a standard tiling argument, that the modal logic with propositional quantifiers and the difference operator is undecidable. Although this fact is already documented in the literature, we believe that the proof is interesting in its own right.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I would like to thank my parents and grandparents for their continued support throughout my academic career, both morally and financially. I would like to thank my supervisor Renate Schmidt for all her help and guidance while working on this dissertation and Jeff Paris for first introducing me to the world of modal logics.

I would further, like to extend my gratitude to those whose pastoral support over the years has been motivational, in particular Carolyn Dean and Marcus Tressl.

Lastly, to the friends who have made studying at Manchester a pleasure; I couldn't have done it without you, thank you.

Chapter 1

Introduction

This dissertation is an exposé of modal logic; an extension of propositional logic by unary operators $\Diamond$ and $\Box$, called modalities, that are not truth-functional. That is to say, the truth or falsity of a modal formula involving modalities is not solely dependent on the truth or falsity of its constituent subformulas.

Modal logics have been studied since the early 1900s and the field of research is now extremely diverse, with applications to natural language processing, commonsense reasoning and modelling knowledge transfer. Our area of interest is Correspondence Theory. The semantics of modal logics is defined on so-called Kripke frames which allow us to express the non-truth-functionality of our modal operators. This semantics define truth over relational structures and, as such, modal logics have a very natural translation into second-order logic, with propositional variables corresponding to second-order variables. However, what is of more interest is that some modal axioms correspond to first-order formulas. For example, we can express the properties of reflexivity, transitivity and euclideanness with relatively simple modal axioms (namely, $\Box p \to p$, $\Box p \to \Box\Box p$ and $\Diamond p \to \Box\Diamond p$ respectively). Sahlqvist, in his 1975 paper, describes a syntactic class of modal axioms that have first-order properties [21]. Moreover, these correspondence properties can be effectively computed. The method he describes for computing such correspondence properties is studied and adapted to a larger class of formulas by van Benthem [24, 25]. The emphasis on there being an effective procedure for computing such correspondence properties is nontrivial, for Chagrova demonstrated that it is undecidable where an arbitrary modal axiom has a first-order correspondence property [6]. That is to say, there is no procedure that can successfully determine the correspondence property for an arbitrary modal formula or terminate in failure if no such correspondent exists.

The Sahlqvist-van Benthem method, therefore, cannot be extended to be able to successfully compute correspondence properties for the whole of modal logic, however, contemporary approaches to the problem are perhaps more successful. Ackermann's Lemma gives us a tool for eliminating second-order quantifiers from formulas of second-order logic that have a particular form [2]. Contemporary algorithms, such as SQEMA, DLS and MSQEL, attempt to manipulate an arbitrary input formula, by a sequence of rewrite rules, until we have a form where we can apply Ackermann's Lemma. Of course, we cannot be sure that failure of these algorithms means lack of first-order correspondent but their correctness ensures that, if a first-order formula is computed as a result of the algorithm, then this is, indeed, the required correspondence property.

We discuss two contemporary algorithms in this dissertation. The first is the Modal Ackermann (MA) calculus, which manipulates sets of modal clauses until we are in a position to apply the modal version of Ackermann's Lemma [23]. The second is a second-order adaptation of the MA-calculus by Schmidt [22], which acts directly on sets of second-order clauses.

The first aim of this dissertation is to study the connection between the MA-calculus, the SOQE-calculus and the traditional method of Sahlqvist and van Benthem. We introduce the necessary background of these various logics and the field of correspondence theory and proceed to describe the three methods for computing correspondence properties. We show that the SOQE-calculus is truly a second-order version of the MA-calculus, in that every rule of the MA-calculus can be emulated by a similar sequence of rules of the SOQE-calculus operating on the standard translation. We discuss the differences in computational power between these contemporary algorithms and the Sahlqvist-van Benthem method. Having highlighted the relationships between these three algorithms, we work towards extending their computational capabilities with various generalizations and extensions.

The second aim is to consider the somewhat famous speech given by former US defense secretary Donald Rumsfeld regarding his doctrine on knowledge and ‘unknown unknowns’. We formalize Rumsfeld’s conception of ‘unknown unkowns’ in the language of modal logic with propositional quantifiers. Only a handful of authors appear to have studied such propositional quantifiers in any depth and the results in the readily available literature are sparse. However, there is general consensus on how the syntax and semantics for second-order modal logics should be formulated (although often in many guises). We describe a class of formulas of this extended language, for which we can effectively compute first-order correspondence properties. We show that our formalism of ‘unknown unknowns’ has a rather interesting first-order correspondence property, namely that of non-euclideanness; indeed, a property that cannot be described by any modal formula of the basic modal language. We remark on the similarities between the modal language with propositional quantifiers, $\mathscr{L}^{\exists}(\Diamond)$, and the modal language with the difference operator, $\mathscr{L}(\Diamond, D)$. We show, via an invariance argument, that although similar, the expressive power of $\mathscr{L}^{\exists}(\Diamond)$ is distinct from that of $\mathscr{L}(\Diamond, D)$.

We consider the combined modal language of propositional quantifiers together with the difference operator $\mathscr{L}^{\exists}(\Diamond, D)$ and show, via a standard tiling argument, that this language is undecidable. Although this fact is nothing new - it having already been documented in the literature that $\mathscr{L}^{\exists}(\Diamond)$ has the same expressive power as second-order logic and is, hence, undecidable already - we believe that the proof is interesting in its own right.

In Chapter 2, we introduce the syntax and semantics for first and second-order logic. Our approach is less traditional and has its roots in the semantics for higher-order logic as described in [26]. The reason for this is to highlight similarities, not only between these two logics but also with modal logic. We also introduce, in this chapter, the syntax and semantics for the first-order logic of fixed-points, as this will be necessary for the generalizations we will make in Chapter 6.

Chapter 3 describes the syntax and semantics for a variety of modal logics. Our main focus will be on the basic modal language with a single modal operator $\Diamond$ and its dual $\Box$. However, we introduce the difference operator D (and its dual $\overline{D}$), together with the global and graded modalities; all of which cannot be expressed, as will we see, in the basic modal language.

We introduce the MA-calculus, the SOQE-calculus and the Sahlqvist-van Benthem method, in Chapter 4, along with some examples. And, in Chapter 5, we describe some of the similarities and differences between the three algorithms. By analyzing the relationships between the algorithms, we are able to draw together some extensions that increase the computational capabilities of all three methods. This is the content of Chapter 6.

We examine Rumsfeld's speech in Chapter 7 and introduce the necessary syntax and semantics for modal logic with propositional quantifiers. We compare this logic with the modal logic $\mathscr{L}(\Diamond, D)$ containing the difference operator and observe that the expressive power of these two logics is, indeed, different. Finally, we discuss undecidability and why Rumsfeld's doctrine on unknown unknowns makes for a poor theory of knowledge.

We conclude what we have accomplished in Chapter 8 and, in Appendix A, we prove, by the tools disscussed in this dissertation, some non-standard correspondence properties which we state in the main body of the dissertation. These examples are also of interest in their own right but stand aside from the natural flow of the dissertation.

Chapter 2

First and Second-Order Logics

2.1 First-Order Logic

First-order logic provides a great deal of expressivity, with which much of mathematics can be formulated. The language of first-order logic (with equality), $\mathscr{L}_1(\tau)$, consists of the symbols $\neg$ ('negation'), $\wedge$ ('conjunction'), $\exists$ ('the existential quantifier'), $\approx$ ('equality') and $\perp$ ('contradiction'), together with a countably infinite set of *first-order variables* (also referred to as *individual variables*). Joined with these is a *similarity type*, $\tau = (\mathscr{R}, \mathscr{C})$, consisting of a set of *second-order constants symbols* $\mathscr{R}$ (also referred to as *relational* or *predicate symbols*) and a set of *first-order constant symbols* $\mathscr{C}$ (referred to, also, as *individual constant symbols*). Assigned to each $R \in \mathscr{R}$ is a natural number n called the *arity* of R.

We define a *first-order term* to be any first-order constant or first-order variable.

Definition 2.1.1. *Atomic formulas* of the language $\mathscr{L}_1(\tau)$ are of the form

$$\alpha ::= R(t_1, \ldots, t_n) \mid s_1 \approx s_2 ,$$

where R is a second-order constant symbol of $\mathscr{R}$ of arity n and each of the t_i are first-order terms, and s_1 and s_2 are also first-order terms.

Formulas of language $\mathscr{L}_1(\tau)$ are formed according to the rule

$$\varphi ::= \alpha \mid \perp \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid (\forall x \ \varphi) ,$$

where α is an atomic formula and x is a first-order variable.

We also make use of abbreviations in Table 2.1 and omit brackets where there is no ambiguity; in particular, we omit the outermost brackets and the brackets between successive conjunctions and disjunctions which are assumed to be associative and commutative. We refer to the symbols $\bot$ and $\top$ as *propositional constants.*

name	*abbreviation*	*equivalent formula*
tautology	$\top$	$\neg \bot$
non-equality	$x \not\approx y$	$\neg(x \approx y)$
disjunction	$(\varphi \vee \psi)$	$\neg(\neg\varphi \wedge \neg\psi)$
implication	$(\varphi \to \psi)$	$(\neg\varphi \vee \psi)$
equivalence	$(\varphi \leftrightarrow \psi)$	$\Big((\varphi \to \psi) \wedge (\psi \to \varphi)\Big)$
universal quantification	$(\forall x \ \varphi)$	$\neg(\exists x \ \neg\varphi)$
unique quantification	$(\exists! x \ \varphi)$	$\Big(\exists y \forall x \ (\varphi \leftrightarrow y \approx x)\Big)$
		(where y does not occur in φ)

Table 2.1: Abbreviations for $\mathscr{L}_1(\tau)$ for all similarity types τ.

With the syntax of $\mathscr{L}_1(\tau)$ as described above, we move now to defining the semantic definition of truth in $\mathscr{L}_1(\tau)$; for this we need the notion of an $\mathscr{L}_1(\tau)$-structure. We follow a slightly less conventional, but equivalent, route to defining semantic truth stemming from the generality of higher-order logics as described in [26]. The reason for this is to push emphasis on the connection between second-order logics and modal logics which we describe in Chapter 3.

Definition 2.1.2. An $\mathscr{L}_1(\tau)$*-structure* $\mathcal{S}$ is a tuple $(W, *)$, where W is a non-empty set called the *universe* or *domain* of $\mathscr{M}$ and $*$ is a function mapping first-order constant symbols of $\mathscr{C}$ to elements of W and second-order constants symbols of $\mathscr{R}$ of arity n to elements of W^n.

A *valuation* on $\mathcal{S}$ is a function V mapping first-order variables to elements of W.

A *model* for $\mathscr{L}_1(\tau)$ with the underlying structure $\mathcal{S}$ is a tuple $(\mathcal{S}, V)$, where V is a valuation on $\mathcal{S}$.

For a valuation V, we have the notion of an *x-equivalent valuation* $V\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)$ which differs from V only at x, where it takes the value $a \in W$. Or more formally:

$$V\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)(y) = \begin{cases} V(y) & \text{if } x \neq y \\ a & \text{if } x = y \end{cases}.$$

Given a model $\mathscr{M} = (\mathcal{S}, V)$ and an x-equivalent valuation $V\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)$ we construct an *x-equivalent model* $\mathscr{M}\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)$ with the same underlying structure $\mathcal{S}$ but with the valuation $V\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)$.

For a first-order term t, we define $t^{\mathscr{M}}$ to be $*(t)$ if t is a constant and $V(t)$ if t is a variable, and for a second-order constant R, define $R^{\mathscr{M}}$ to be $*(R)$.

Given a model $\mathscr{M}$ for $\mathscr{L}_1(\tau)$ and a formula φ of $\mathscr{L}_1(\tau)$ we define semantic truth inductively on the length of φ as follows:

$$\begin{aligned}
\mathscr{M} \models R(t_1, \ldots, t_n) &\iff (t_1^{\mathscr{M}}, \ldots, t_n^{\mathscr{M}}) \in R^{\mathscr{M}} \text{ if } R \text{ is a second-order constant of arity } n, \\
\mathscr{M} \models t_1 \approx t_2 &\iff t_1^{\mathscr{M}} = t_2^{\mathscr{M}} \qquad \text{if } t_1, t_2 \text{ are first-order terms}, \\
\mathscr{M} \models \neg\varphi &\iff \mathscr{M} \not\models \varphi\ , \\
\mathscr{M} \models (\varphi \wedge \psi) &\iff \mathscr{M} \models \varphi \text{ and } \mathscr{M} \models \psi\ , \\
\mathscr{M} \models (\exists x\ \varphi) &\iff \mathscr{M}' \models \varphi \text{ for some } x\text{-equivalent model } \mathscr{M}'.
\end{aligned}$$

Each of the t_i are first-order terms, x is a first-order variable and φ, ψ are themselves formulas of $\mathscr{L}_1(\tau)$. We note that under this semantics the operations of conjunction and, consequently, disjunction are commutative and associative, as we require.

We say that a first-order formula, φ, is *valid* on a structure $\mathcal{S}$ if $\mathscr{N} \models \varphi$ for all models with the underlying structure $\mathcal{S}$.

First-order logics satisfies a nice property regarding the cardinality of their models, which we quote here.

Theorem 2.1.1 (Skolem-Löwenheim)**.** *Given a set of $\mathscr{L}_1(\tau)$-sentences, Σ, and a* *structure $\mathcal{S}$ such that $\mathcal{S} \models \varphi$ for all $\varphi \in \Sigma$ then there is a structure $\mathcal{S}'$ of arbitrary cardinality $\kappa \geq \aleph_0$ such that $\mathcal{S}' \models \varphi$ for all $\varphi \in \Sigma$.*

However, for similarity types τ, containing at least one second-order constants of arity greater than 1, the satisfiability problem for $\mathscr{L}_1(\tau)$ is undecidable [15].

A language is said to be *undecidable* if the satisfiability problem for that language is undecidable; that is to say there is no decision procedure for effectively deducing whether, for a given formula φ, there is some model $\mathscr{N}$ such that $\mathscr{N} \models \varphi$.

We now turn our attention toward second-order logic as a natural extension of first-order logic.

2.2 Second-Order Logic

Second-Order Logic is the extension of first-order logic by quantification over predicate symbols as well as the usual quantification over individuals. It allows us to formulate sentences of the form

$$\exists P \ \forall Q \ \forall x \ \Big(Px \to Qx\Big) \ ,$$

which can be interpreted as saying that there is some predicate P which contains every other predicate Q. However, we must formalize what we mean be quantification over all / some predicates.

As with the first-order language $\mathscr{L}_1(\tau)$, the language of second-order logic (with equality), $\mathscr{L}_2(\tau)$, consists of the symbols $\neg$, $\wedge$, $\exists$, $\approx$, $\perp$, and a countably infinite set of first-order variables. However, we also have an additional countably infinite set of *second-order variables* of arity n for each $n \in \mathbb{N}$. Following [10] we take the convention of using boldface uppercase Roman characters to denote second-order variables. The similarity type $\tau = (\mathscr{R}, \mathscr{C})$ is defined as it was for first-order logic.

We define a first-order (resp. second-order) *term* to be any first-order (resp. second-order) constant or variable.

Definition 2.2.1. *Atomic formulas* of the language $\mathscr{L}_2(\tau)$ are of the form

$$\alpha ::= R(t_1, \ldots, t_n) \;\;|\;\; \mathbf{P}(t_1, \ldots, t_n) \;\;|\;\; t_1 \approx t_2 \;\;|\;\; S_1 \approx S_2 \;,$$

where R is a second-order constant symbol of $\mathscr{R}$ of arity n, $\mathbf{P}$ is a second-order variable of arity n, the t_i are first-order terms, and the S_i are second-order terms.

Formulas of the language $\mathscr{L}_2(\tau)$ are formed according to the rule

$$\varphi ::= \alpha \;\;|\; \bot \;|\;\; \neg\varphi \;\;|\;\; (\varphi_1 \wedge \varphi_2) \;\;|\;\; (\exists x \;\, \varphi) \;\;|\;\; (\exists \mathbf{P} \;\, \varphi) \;\;,$$

where α is an atomic formula, x is a first-order variable and $\mathbf{P}$ is a second-order variable.

We also make use of the same abbreviations given in Table 2.1 as well as the additional abbreviations given below in Table 2.2.

name	*abbreviation*	*equivalent formula*
universal quantification	$(\forall \mathbf{P} \;\, \varphi)$	$\neg\,(\exists \mathbf{P} \;\, \neg\varphi)$
unique quantification	$(\exists! \mathbf{P} \;\, \varphi)$	$\Big(\exists \mathbf{Q} \forall \mathbf{P} \;\, (\varphi \leftrightarrow \mathbf{P} \approx \mathbf{Q})\Big)$
		(where $\boldsymbol{Q}$ *does not occur in* φ*)*

Table 2.2: Additional abbreviation for $\mathscr{L}_2(\tau)$ for all similarity types τ.

We now define the notion of the *polarity* of a formula of $\mathscr{L}_2(\tau)$. Let $PNF(\varphi)$ be the prenex normal form of φ, which can be effectively computed using the rules of Table 2.3. A modal formula φ is *positive with respect to a second-order variable* $\boldsymbol{P}$ if every occurrence of $\mathbf{P}$ in the quantifier-free part of $PNF(\varphi)$ is in the scope of an *even* number of negation signs and *negative (w.r.t* $\boldsymbol{P}$*)* if every occurrence of $\mathbf{P}$ in the quantifier-free part of $PNF(\varphi)$ is in the scope of an *odd* number of negation signs.

A formula is *positive* (resp. *negative*) if it is positive (resp. negative) with respect to all propositional variables that occur in it.

input formula	*equivalent formula*
$\neg(\varphi \leftrightarrow \psi)$	$(\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
$\varphi \leftrightarrow \psi$	$(\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$
$\neg(\varphi \rightarrow \psi)$	$\varphi \wedge \neg\psi$
$\varphi \rightarrow \psi$	$\neg\varphi \vee \psi$
$\neg(\varphi \vee \psi)$	$\neg\varphi \wedge \neg\psi$
$\neg(\varphi \wedge \psi)$	$\neg\varphi \vee \neg\psi$
$\neg\neg\varphi$	φ
$\neg\perp$	$\top$
$\neg\top$	$\perp$
$\neg\forall x\ \varphi$	$\exists x\ \neg\varphi$
$\neg\exists x\ \varphi$	$\forall x\ \neg\varphi$
$\forall x[\alpha(x)] \wedge \forall y[\beta(y)]$	$\forall z[\alpha(z) \wedge \beta(z)]$
$\exists x[\alpha(x)] \vee \exists y[\beta(y)]$	$\exists z[\alpha(z) \vee \beta(z)]$
$\alpha \vee Qx[\beta(x)]$	$Qz[\alpha \vee \beta(z)]$
$\alpha \wedge Qx[\beta(x)]$	$Qz[\alpha \wedge \beta(z)]$
$Qx[\alpha(x)] \vee \beta$	$Qz[\alpha(z) \vee \beta]$
$Qx[\alpha(x)] \wedge \beta$	$Qz[\alpha(z) \wedge \beta]$
where $Q \in \{\forall, \exists\}$ and z is a fresh variable	

Table 2.3: Rules for computing the prenex pormal form

We now define $\mathscr{L}_2(\tau)$-structures, $\mathscr{L}_2(\tau)$-models and semantic truth in $\mathscr{L}_2(\tau)$, analogous to the definitions for $\mathscr{L}_1(\tau)$ given in Definition 2.1.2.

Definition 2.2.2. An $\mathscr{L}_2(\tau)$*-structure* $\mathcal{S}$ is a tuple $(W, *)$, where, as before, W is a non-empty set called the *universe* or *domain* of $\mathcal{S}$ and $*$ is a function mapping first-order constant symbols of $\mathscr{C}$ to elements of W and second-order constants symbols of $\mathscr{R}$ of arity n to elements of W^n.

A *valuation* on $\mathcal{S}$ is a function V mapping first-order variables to elements of W and second-order variables of arity n to elements of W^n.

A *model* for $\mathscr{L}_2(\tau)$ with the underlying structure $\mathcal{S}$ is a tuple $\mathscr{N} = (\mathcal{S}, V)$ where V is a valuation on $\mathcal{S}$.

In addition to the notion of an *x-equivalent valuation* $V\left(\begin{smallmatrix} x \\ a \end{smallmatrix}\right)$ defined above, we have the notion of a ***P**-equivalent valuation* $V\left(\begin{smallmatrix} \mathbf{P} \\ A \end{smallmatrix}\right)$ which differs from V only at $\mathbf{P}$ where it takes the value $A \in \mathcal{P}\left(W^n\right)$ where $\mathbf{P}$ is a second-order variable of arity n, and $\mathcal{P}\left(X\right)$ denotes the powerset of the set X. More formally,

$$V\left(\begin{smallmatrix} \mathbf{P} \\ A \end{smallmatrix}\right)(\mathbf{Q}) = \begin{cases} V(y) & \text{if } \mathbf{Q} \neq \mathbf{P} \\ A & \text{if } \mathbf{Q} = \mathbf{P} \end{cases} .$$

A ***P**-equivalent model* is a model with the same underlying structure but with a $\mathbf{P}$-equivalent valuation. We write $\mathscr{N}\left(\begin{smallmatrix} \mathbf{P} \\ A \end{smallmatrix}\right)$ for the ***P**-equivalent model* with valuation $V\left(\begin{smallmatrix} \mathbf{P} \\ A \end{smallmatrix}\right)$.

For any term t, we define $t^{\mathscr{N}}$ to be $*(t)$ if t is a constant and $V(t)$ if t is a variable.

Now, given a model $\mathscr{N}$ for $\mathscr{L}_2(\tau)$ and a formula φ of $\mathscr{L}_2(\tau)$ we define semantic truth inductively on the length of φ as follows:

$$\begin{aligned}
\mathscr{N} \models R(t_1, \ldots, t_n) &\iff (t_1^{\mathscr{N}}, \ldots, t_n^{\mathscr{N}}) \in R^{\mathscr{N}} \text{ if } R \text{ is a second-order constant of arity } n, \\
\mathscr{N} \models \mathbf{P}(t_1, \ldots, t_n) &\iff (t_1^{\mathscr{N}}, \ldots, t_n^{\mathscr{N}}) \in \mathbf{P}^{\mathscr{N}} \text{ if } \mathbf{P} \text{ is a second-order variable of arity } n, \\
\mathscr{N} \models s_1 \approx s_2 &\iff s_1^{\mathscr{N}} = s_2^{\mathscr{N}} \qquad \text{if } s_1, s_2 \text{ are terms of the same order}, \\
\mathscr{N} \models \neg\varphi &\iff \mathscr{N} \not\models \varphi \;, \\
\mathscr{N} \models (\varphi \wedge \psi) &\iff \mathscr{N} \models \varphi \text{ and } \mathscr{M} \models \psi \;, \\
\mathscr{N} \models (\exists x \;\; \varphi) &\iff \mathscr{N}' \models \varphi \text{ for some } x\text{-equivalent model } \mathscr{N}', \\
\mathscr{N} \models (\exists \mathbf{Q} \;\; \varphi) &\iff \mathscr{N}' \models \varphi \text{ for some } \mathbf{Q}\text{-equivalent model } \mathscr{N}',
\end{aligned}$$

where each of the t_is are first-order terms, x and $\mathbf{Q}$ are first and second-order variables respectively and φ, ψ are, themselves, formulas of $\mathscr{L}_1(\tau)$.

We say that a formula φ is *(upwards-)monotone* with respect to a second-order variable $\mathbf{P}$ if for all $\mathbf{P}$-equivalent models $\mathscr{N}_1 = (W, *, V_1)$ and $\mathscr{N}_2 = (W, *, V_2)$ such that $V_1(\mathbf{P}) \subseteq V_2(\mathbf{P})$, we have that $\mathscr{N}_1 \models \varphi \Rightarrow \mathscr{N}_2 \models \varphi$.

We say that φ is *downwards-monotone* with respect to $\mathbf{P}$ if for all $\mathbf{P}$-equivalent models $\mathscr{N}_1 = (W, *, V_1)$ and $\mathscr{N}_2 = (W, *, V_2)$ such that $V_1(\mathbf{P}) \subseteq V_2(\mathbf{P})$, we have that $\mathscr{N}_1 \models \varphi \Leftarrow \mathscr{N}_2 \models \varphi$.

This definition of monotonicity of a formula is purely *semantic* in nature, whereas the definition of polarity of a formula was purely *syntactic*, however, the following result describes a nice connection between the two; one which we will heavily rely on in later chapters.

Proposition 2.2.1. *If φ is positive with respect to some second-order variable $\boldsymbol{P}$ then φ is (upwards-)monotone with respect to $\boldsymbol{P}$ and if φ is negative with respect to $\boldsymbol{P}$ then φ is downwards-monotone with respect to $\boldsymbol{P}$.*

Proof: Not given. See [13], page 27. ∎

It is worth noting that although we have included equality as a primitive symbols of $\mathscr{L}_2$, we could have achieved the same level of expressivity had we omitted $\approx$ from our language. We would then use $\approx$ as a *defined* symbol, making use of the abbreviations of Table 2.4 [19].

name	*abbreviation*	*equivalent formula*
first-order equality	$t_1 \approx t_2$	$\forall \mathbf{P}\Big(P(t_1) \leftrightarrow P(t_2)\Big)$
second-order equality	$S_1 \approx S_2$	$\forall x_1 \ldots \forall x_n \Big(S_1(x_1, \ldots, x_n) \leftrightarrow S_2(x_1, \ldots, x_n)\Big)$
where the t_i are first-order terms and the S_i are second-order terms of arity n.		

Table 2.4: Equality abbreviations for second-order logic.

It is easily checked that, unfolding the semantic definition of these abbreviations, we get precisely the same definition we gave for $\approx$ as a primitive symbol of $\mathscr{L}_2(\tau)$.

We now have a formal grounding for talking about quantification over predicates, but do we really need such extensions of first-order logic? Can we not already say everything we could hope to say in some first-order language? We see that this cannot be the case and that second-order logic is strictly more expressive than any first-order logic. In particular, formulas of second-order logic can make assertions about the cardinality of their models, whereas formulas of first-order logic cannot. Consequently, second-order logic violates the Skolem-Löwenheim Theorem.

Proposition 2.2.2. *Second-order logic violates the Skolem-Löwenheim theorem.*

Proof: We assume for contradiction that the Skolem-Löwenheim theorem holds for second-order logic. Consider the following second-order formulas

$$\text{function}(\mathbf{F},\mathbf{X},\mathbf{Y}) \iff \forall x \exists y \Big[\mathbf{F}(x,y) \rightarrow \Big(\mathbf{X}(x) \wedge \mathbf{Y}(y) \wedge \forall z \left[\mathbf{F}(x,z) \rightarrow z \approx y \right] \Big) \Big] \ ,$$

$$\begin{aligned} &\text{injection}(\mathbf{F},\mathbf{X},\mathbf{Y}) \iff \text{function}(\mathbf{F},\mathbf{X},\mathbf{Y}) \\ &\wedge \forall x_1 \forall x_2 \Big[(\mathbf{X}(x_1) \wedge \mathbf{X}(x_2) \wedge x_1 \not\approx x_2) \rightarrow \forall y \Big(\mathbf{Y}(y) \rightarrow \neg \mathbf{F}(x_1,y) \vee \neg \mathbf{F}(x_2,y) \Big) \Big] \ , \end{aligned}$$

and

$$\text{surjection}(\mathbf{F},\mathbf{X},\mathbf{Y}) \iff \text{function}(\mathbf{F},\mathbf{X},\mathbf{Y}) \wedge \forall y \exists x \Big[\mathbf{Y}(y) \wedge \mathbf{X}(x) \wedge \mathbf{F}(x,y) \Big] \ .$$

These formulas define respectively, the properties that the relation $\mathbf{F}$ is a function, is an injection and is a surjection between the sets $\mathbf{X}$ and $\mathbf{Y}$, respectively.

We can now define the properties that an unary predicate $\mathbf{P}$ is finite with the formula finite($\mathbf{P}$) which says that every surjective function from $\mathbf{P}$ to itself is injective:

$$\text{finite}(\mathbf{P}) \iff \forall \mathbf{F} \Big[\text{surjection}(\mathbf{F},\mathbf{P},\mathbf{P}) \rightarrow \text{injection}(\mathbf{F},\mathbf{P},\mathbf{P}) \Big] \ ,$$

and the property that $\mathbf{P}$ is countably infinite with the formula countable($\mathbf{P}$) which says that for every infinite subset $\mathbf{Q}$ of $\mathbf{P}$, there is a bijection, $\mathbf{F}$ between $\mathbf{Q}$ and $\mathbf{P}$:

$$\begin{aligned} \text{countable}(\mathbf{P}) \iff \forall \mathbf{Q} \ \Big[&\forall x \Big(\mathbf{P}(x) \rightarrow \mathbf{Q}(x) \Big) \wedge \neg \text{finite}(\mathbf{Q}) \\ &\rightarrow \exists \mathbf{F} \Big(\text{injection}(\mathbf{F},\mathbf{P},\mathbf{Q}) \wedge \text{surjection}(\mathbf{F},\mathbf{P},\mathbf{Q}) \Big) \Big] \ . \end{aligned}$$

Suppose that $\mathcal{S}$ is a structure that satisfies the formula $\varphi := \forall \mathbf{P} \ \text{countable}(\mathbf{P})$ then by the supposed correctness of the Skolem-Löwenheim theorem, φ must be satisfiable on some model of arbitrarily high cardinality $\kappa > \aleph_0$. However, this is a contradiction since φ attests to the fact that the universe of any model is countable.

Hence second-order logic violates the Skolem-Löwenheim theorem. ∎

Since second-order logic is an extension of first-order logic, we have that the satisfiability problem for $\mathscr{L}_2(\tau)$ is, too, undecidable for all similarity types containing at least one second-order constant of arity n. However we can, in fact, show that something stronger; namely that $\mathscr{L}_2(\tau)$ is undecidable for any similarity type τ.

We take this opportunity to introduce the domino problem of [27] as a standard way of proving undecidability results.

Definition 2.2.3. A domino system $\mathcal{D}$ is a tuple (T, H, V) where T is a set of domino types and $H, V \subseteq T \times T$ are sets of horizontal and vertical matching conditions respectively. Intuitively if D_i, D_j are domino types of D and $(D_i, D_j) \in H$ then D_j may be placed horizontally to the right of D_i as illustrated in Figure 2.1 where we have

$$\begin{aligned} H &= \{(D_1, D_4), (D_2, D_5), (D_3, D_1), (D_4, D_5), (D_5, D_1), (D_5, D_2)\} \ , \\ V &= \{(D_1, D_3), (D_2, D_5), (D_3, D_5), (D_4, D_1), \\ &\qquad (D_4, D_2), (D_4, D_4), (D_5, D_1), (D_5, D_2), (D_5, D_4)\} \ . \end{aligned}$$

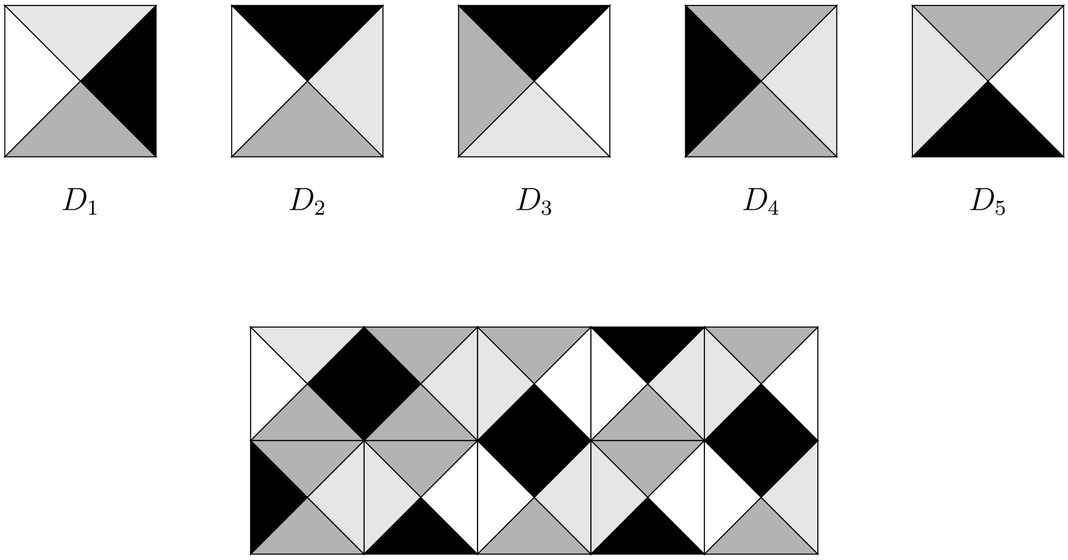


Figure 2.1: Illustration of domino types being tiled

The *domino problem* asks if there is a decision procedure for determining whether a given domino system can tile the $\mathbb{N} \times \mathbb{N}$ plane.

This problem was proved to be undecidable by Berger in his 1966 paper [3] by reducing an arbitrary Turning Machines M, on a one-way infinite tape (that is initially empty) to a domino system $\mathcal{D}(M)$ such that $\mathcal{D}(M)$ tiles the $\mathbb{N} \times \mathbb{N}$ plane if and only if M does not halt.

Since the Halting Problem for the set of Turning Machines on a one-way infinite tape (that is initially empty) is undecidable, so too must the domino problem be undecidable.

We now use this result to show that the second-order logic is undecidable for all similarity types.

Theorem 2.2.1. *The satisfiability problem for $\mathscr{L}_2(\emptyset)$ is undecidable.*

Proof: We prove that the domino problem is reducible to the satisfiability problem of $\mathscr{L}_2(\emptyset)$ which, of course, sits inside $\mathscr{L}_2(\tau)$.

Given a domino system $\mathcal{D} = (T, H, V)$ where $T = \{D_1, \ldots, D_n\}$ is a set of domino types and $H, V \subseteq T \times T$ are the horizontal and vertical conditions.

We associate each D_i of T with an unary second-order variable $\mathbf{D}_i$ and introduce the binary second-order variables $\mathbf{X}$ and $\mathbf{Y}$ which will allow us to impose a grid-like structure on the universe. We now construct the following set of $\mathscr{L}_2(\emptyset)$-formulas:

$$
\begin{aligned}
\theta &= \exists x \top , \\
\varphi_1 &= \forall x\Big(\mathbf{D}_1 x \vee \ldots \vee \mathbf{D}_n x\Big) , \\
\varphi_2 &= \bigwedge_{i=1}^{n-1} \forall x \left[\mathbf{D}_i x \rightarrow \bigwedge_{j=i}^{n-1} \neg \mathbf{D}_{j+1} x\right] , \\
\psi_1 &= \forall x \exists! y \ \mathbf{X}(x,y) , \\
\psi_2 &= \forall x \exists! y \ \mathbf{Y}(x,y) , \\
\psi_3 &= \forall x \forall y \forall z \Big[\Big(\mathbf{X}(x,y) \wedge \mathbf{Y}(x,z)\Big) \rightarrow \exists w \Big(\mathbf{X}(z,w) \wedge Y(y,w)\Big)\Big] , \\
\chi_1 &= \bigwedge_{i=1}^{n} \forall x \forall y \left[\mathbf{D}_i x \wedge \mathbf{X}(x,y) \rightarrow \bigvee_{(D_i,D_j)\in H} \mathbf{D}_j y\right] , \\
\chi_2 &= \bigwedge_{i=1}^{n} \forall x \forall y \left[\mathbf{D}_i x \wedge \mathbf{Y}(x,y) \rightarrow \bigvee_{(D_i,D_j)\in V} \mathbf{D}_j y\right] .
\end{aligned}
$$

Then let Φ be the formula $\Big[\theta \wedge \varphi_1 \wedge \varphi_2 \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \chi_1 \wedge \chi_2\Big]$ then we now show that Φ is satisfiable if and only if the domino system $\mathcal{D}$ tiles the $\mathbb{N} \times \mathbb{N}$ plane.

Suppose that $t : \mathbb{N} \times \mathbb{N} \rightarrow T$ is a tiling of $\mathbb{N} \times \mathbb{N}$ then define the model $\mathscr{N}$ with underlying structure $(\mathbb{N} \times \mathbb{N}, \emptyset)$ and valuation V, where

$$
\begin{aligned}
V(\mathbf{X}) &= \Big\{\Big((n,m),(n+1,m)\Big) \mid n,m \in \mathbb{N}\Big\} , \\
V(\mathbf{Y}) &= \Big\{\Big((n,m),(n,m+1)\Big) \mid n,m \in \mathbb{N}\Big\} , \\
V(\mathbf{D}_i) &= \{(n,m) \mid n,m \in \mathbb{N} \text{ and } t(n,m) = D_i\} .
\end{aligned}
$$

It is straightforward to check that $\mathscr{N}$ is indeed a model for Φ.

Conversely, suppose that $\mathscr{N}$ is a model for Φ then $\mathscr{N} \models \theta \wedge \psi_1 \wedge \psi_2 \wedge \psi_3$, from which we deduce that there is a function $f : \mathbb{N} \times \mathbb{N} \rightarrow W$ such that $f(0,0) = w$ for some $w \in W$, $\Big(f(n,m), f(n+1,m)\Big) \in V(\mathbf{X})$ and

$\Big(f(n,m), f(n,m+1)\Big) \in V(\mathbf{Y})$ for all $n,m \in \mathbb{N}$.

We can then define a tiling $t : \mathbb{N} \times \mathbb{N} \to T$ as follows:

$$t(n,m) = D_i \text{ if and only if } f(n,m) \in V(\mathbf{D}_i) \ .$$

Since $\mathscr{N} \models \varphi_1 \wedge \varphi_2$ the subsets $V(\mathbf{D}_1), \ldots, V(\mathbf{D}_n)$ must form a partition; that is to say $\bigcup_{i=1}^{n} V(\mathbf{D}_1) = W$ and $V(\mathbf{D}_i) \cap V(\mathbf{D}_j) = \emptyset$ whenever $i \neq j$, and hence the tiling t is well-defined.

We note that, since $\mathscr{N} \models \chi_1$, if $f(n,m) \in V(\mathbf{D}_i)$ and $f(n+1,m) \in V(\mathbf{D}_j)$ then $(\mathbf{D}_i, \mathbf{D}_j) \in H$. And also, since $\mathscr{N} \models \chi_2$, if $f(n,m) \in V(\mathbf{D}_i)$ and $f(n,m+1) \in V(\mathbf{D}_j)$ then $(\mathbf{D}_i, \mathbf{D}_j) \in V$.

Hence we have that since the domino problem is undecidable, and is effectively reducible to a satisfiability problem in $\mathscr{L}_2(\emptyset)$, so too must the satisfiability problem of $\mathscr{L}_2(\emptyset)$ be undecidable. ∎

Consequently, the satisfiability of $\mathscr{L}_2(\tau)$ is undecidable for any similarity type τ.

2.3 The First-Order Logic of Fixed-Points

We now introduce an extension to first-order logic which we will use in Chapter 4, namely first-order logic of fixed-points as defined in [13] (pages 33–35). Our route to this logic will go via second-order logic of fixed-points, the first-order logic of fixed-points being considered as a restriction of this larger class of formulas.

Definition 2.3.1. Let $\mathscr{L}_2^{\mu}(\tau)$ be the extension of the language $\mathscr{L}_2(\tau)$ by the symbol LFP ('least fixed-point') and construct atomic formulas according to the rule:

$$\alpha ::= R(t_1, \ldots, t_n) \ \mid\ \mathbf{P}(t_1, \ldots, t_n) \ \mid\ t_1 \approx t_2 \ \mid\ [\textsc{lfp}\ \mathbf{P}.\ \varphi(\mathbf{P})]\,(t_1, \ldots, t_n)\ .$$

Here, R is a second-order constant of arity n, $\mathbf{P}$ is a second-order variable of arity n, and the t_i are first-order terms. Also, φ is any formula of $\mathscr{L}_2^{\mu}(\tau)$ in which $\mathbf{P}$ occurs freely and positively. We say that the variable $\mathbf{P}$, if it occurs in φ, occurs *bound* by the least fixed-point construct.

Formulas of $\mathscr{L}_2^{\mu}(\tau)$ are constructed according to the following rule

$$\varphi ::= \alpha \ \mid \bot \mid\ \neg\varphi \ \mid\ (\varphi_1 \wedge \varphi_2) \ \mid\ (\exists x\ \varphi) \ \mid\ (\exists \mathbf{P}\ \varphi)\ ,$$

where α is an atomic formula and $\mathbf{P}$ is a second-order variable.

We make use of the additional abbreviation in Table 2.5 for 'greatest fixed-point':

abbreviation	*equivalent formula*
$[\textsc{gfp}\ \mathbf{P}.\ \varphi(\mathbf{P})]\,(\vec{t})$	$\neg\left[\textsc{lfp}\ \mathbf{P}.\ \neg\varphi(\mathbf{P})\left\{{}^{\mathbf{P}(\vec{u})}/_{\neg\mathbf{P}(\vec{u})}\right\}\right](\vec{t})$

Table 2.5: Additional abbreviation for $\mathscr{L}_2^{\mu}(\tau)$ for all similarity types τ.

To describe the semantics of $\mathscr{L}_2^{\mu}(\tau)$ we first need to define the *lfp-sequence* of ordinal approximations for a positive formula φ with respect to $\mathbf{P}$ in some model $\mathscr{N}$. Given a model $\mathscr{N}$, let

$$F_{\varphi,\mathscr{N}}(X) = \left\{\vec{w} \in X \ \mid\ \mathscr{N}\left(\begin{smallmatrix}\mathbf{P}\\ X\end{smallmatrix}\right)\left(\begin{smallmatrix}\vec{x}\\ \vec{w}\end{smallmatrix}\right) \models \varphi(\vec{x})\right\}\ ,$$

where $\mathscr{N}\left(\begin{smallmatrix}\mathbf{P}\\ X\end{smallmatrix}\right)\left(\begin{smallmatrix}\vec{x}\\ \vec{w}\end{smallmatrix}\right)$ is the $(\mathbf{P},\vec{x})$-equivalent model with the valuation that maps $\mathbf{P}$ to the set X and each of the free variables of $\vec{x}$ to the elements $\vec{w}$.

The sequence $(\varphi^{\alpha}_{\mathscr{N}} \mid \alpha \geq 0)$ given by the rule

$$\begin{aligned}
\varphi^{0}_{\mathscr{N}} &= \emptyset \ , \\
\varphi^{\alpha+1}_{\mathscr{N}} &= F_{\varphi,\mathscr{N}}(\varphi^{\alpha}_{\mathscr{N}}) \ , \\
\varphi^{\lambda}_{\mathscr{N}} &= \bigcup_{\alpha<\lambda} \varphi^{\alpha}_{\mathscr{N}} \ ,
\end{aligned}$$

is then monotonically increasing and, due to the Tarski-Knaster Theorem (see [25]), stabilizes for some ordinal number ζ, which we call the *closure ordinal* [13] (page 35).

We write,

$$[\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}} = \varphi^{\zeta}_{\mathscr{N}} \ .$$

The semantics of $\mathscr{L}_2(\tau)$ are, then, extended to formulas of $\mathscr{L}_2^{\mu}(\tau)$ by providing the rule

$$\mathscr{N} \models [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]\,(t_1,\ldots,t_2) \iff (t_1^{\mathscr{N}},\ldots,t_n^{\mathscr{N}}) \in [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}} \ .$$

We have the following proposition which will be crucial to the algorithms we introduce in Chapter 4. We believe the result is standard but were unable to find an existing proof in the literature.

Proposition 2.3.1. *If* $[\text{LFP } \boldsymbol{P}.\ \varphi(\boldsymbol{P})]\,(t_1,\ldots,t_n)$ *is positive with respect to* $\boldsymbol{Q}$ *then*
$[\text{LFP } \boldsymbol{P}.\ \varphi(\boldsymbol{P})]\,(t_1,\ldots,t_n)$ *is upwards-monotone with respect to* $\boldsymbol{Q}$*.*

Proof: Suppose that $[\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]\,(t_1,\ldots,t_n)$ is positive with respect to $\mathbf{Q}$, then we must have that φ is positive with respect to $\mathbf{Q}$.

We consider the lfp-sequences, $\varphi^{\alpha}_{\mathscr{N}_1}$ and $\varphi^{\alpha}_{\mathscr{N}_2}$, for φ with respect to $\mathbf{P}$, for the $\mathbf{Q}$-equivalent models $\mathscr{N}_1$ and $\mathscr{N}_2$ such that $V_1(\mathbf{Q}) \subseteq V_2(\mathbf{Q})$.

Trivially we have that $\varphi^{0}_{\mathscr{N}_1} \subseteq \varphi^{0}_{\mathscr{N}_2} = \emptyset$, and by the definition, since φ is positive, and hence monotone with respect to all second-order variable, we have that

$$F_{\varphi,\mathscr{N}_1}(X) = \left\{\vec{s} \in X \ \mid \ \mathscr{N}_1\left(\tfrac{\vec{x}}{\vec{s}}\right) \models \varphi(\vec{x})\right\} \subseteq \left\{\vec{s} \in X \ \mid \ \mathscr{N}_2\left(\tfrac{\vec{x}}{\vec{s}}\right) \models \varphi(\vec{x})\right\} = F_{\varphi,\mathscr{N}_2}(X)$$

and, consequently, $\varphi^{\alpha}_{\mathscr{N}_1} \subseteq \varphi^{\alpha}_{\mathscr{N}_2}$ for all ordinal numbers α.

Let ζ_1 and ζ_2 be the closure ordinals of the lfp-sequences $\varphi^{\alpha}_{\mathscr{N}_1}$ and $\varphi^{\alpha}_{\mathscr{N}_2}$ respectively and let $\zeta = \max(\zeta_1, \zeta_2)$. We have, then, that

$$[\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}_1} = \varphi^{\zeta}_{\mathscr{N}_1} \subseteq \varphi^{\zeta}_{\mathscr{N}_2} = [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}_2} \ ,$$

from which, it follows that

$$\begin{aligned} \mathscr{N}_1 \models [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]\,(t_1,\ldots,t_n) &\iff (t_1^{\mathscr{N}_1},\ldots,t_n^{\mathscr{N}_1}) \in [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}_1} \\ &\implies (t_1^{\mathscr{N}_2},\ldots,t_n^{\mathscr{N}_2}) \in [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]^{\mathscr{N}_2} \\ &\iff \mathscr{N}_1 \models [\text{LFP } \mathbf{P}.\ \varphi(\mathbf{P})]\,(t_1,\ldots,t_n)\ , \end{aligned}$$

since $t_i^{\mathscr{N}_1} = t_i^{\mathscr{N}_2}$ for each $1 \le i \le n$. This is precisely how we defined monotonicity, so we are done. ∎

The fragment of this logic we will be interested in is the *first-order logic of fixed-points*, $\mathscr{L}_1^{\mu}(\Diamond)$, which is defined as the restriction of the set of $\mathscr{L}_2^{\mu}(\Diamond)$-formula, to formulas which contain no free occurrences of second-order variables and contain no subformulas of the form $(\exists\mathbf{P}\ \psi)$. That is to say, all occurrences of second-order variables are bound by the fixed-point constructor LFP.

We now move away from second-order logic and consider modal logics; a modest extension of propositional logic by the unary symbol $\Diamond$ whose semantic definition is not truth functional, and thus is a strictly stronger language than propositional logic.

Chapter 3

Modal Logic and Correspondence Theory

3.1 The Basic Modal Logic $\mathscr{L}(\Diamond)$

Modal Logics arise from augmenting propositional logic with modalities which, unlike Boolean connectives, are not truth-functional; that it to say, their truth or falsity is not solely dependent on the truth of falsity of the subformula(s) which they bind. Such modalities are ubiquitous in natural language, for example, of the statements "My brother knows that 17 is prime" and "My brother knows that $2^{43,112,609} - 1$ is prime", the first is true and the second, having put him to the test, is false, however the statements "17 is prime" and "$2^{43,112,609} - 1$ is prime" are both true.

Of course, all my brother need do is run a Google search to find out that $2^{43,112,609} - 1$ is in fact the largest known Mersenne prime (at the time of writing) and the truth of the statement "My brother knows that $2^{43,112,609} - 1$ is prime" would change! This transition from acquiring new knowledge is again not truth-functional. Other examples include statements such as: "I believe that The Riemann Hypothesis is true", which I attest to being a true statement even if I am in fact wrong in that belief, and "I will always remember my 21st birthday", which although may be a true statement, it certainly wasn't a true statement prior to my 21st birthday.

All these cases can be modelled with various modal logics but we will focus predominantly on the basic modal language $\mathscr{L}(\Diamond)$ consisting of the symbols $\neg$ ('negation'), $\wedge$ ('conjunction'), $\Diamond$ ('possibility') and $\bot$ ('contradiction'), together with a countably infinite set of *propositional variables*.

Definition 3.1.1. Modal formulas of the language $\mathscr{L}(\Diamond)$ are given by the rule:

$$\varphi ::= p \mid \bot \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \Diamond\varphi \ ,$$

where p is a propositional variable.

We use the abbreviations in Table 3.1 and again, we omit brackets where there is no ambiguity; in particular, we omit the outermost brackets and the brackets between successive conjunctions and disjunctions. We refer to the symbols $\bot$ and $\top$ as *propositional constants*.

name	*abbreviation*	*equivalent formula*
tautology	$\top$	$\neg\bot$
disjunction	$(\varphi \vee \psi)$	$\neg(\neg\varphi \wedge \neg\psi)$
implication	$(\varphi \rightarrow \psi)$	$(\neg\varphi \vee \psi)$
equivalence	$(\varphi \leftrightarrow \psi)$	$((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$
necessity	$\Box\varphi$	$\neg\Diamond\neg\varphi$

Table 3.1: Abbreviations for $\mathscr{L}(\Diamond)$.

The underlying key to correctly interpreting formulas of the form $\Diamond\varphi$ is the notion of Kripke frames.

Definition 3.1.2. A *Kripke frame* $\mathscr{F}$ is defined to be a pair $(W, \widehat{R})$ where W is a non-empty set and $\widehat{R}$ is a binary relation on W, that is $\widehat{R} \subseteq W \times W$. A *valuation* on $\mathscr{F}$ is a function V mapping each propositional variable p to a subset of W.

Truth of a modal formula is defined inductively over *Kripke models* of the form $\mathscr{M} = (\mathscr{F}, V)$ where $\mathscr{F}$ is a Kripke frame and V is a valuation on $\mathscr{F}$.

We now have for each valuation V, the notion of a *p-equivalent* valuation $V\binom{p}{a}$ which differs from V only at p where it takes the value $a \subseteq W$. Or more formally:

$$V\left(\begin{smallmatrix}p\\a\end{smallmatrix}\right)(q) = \begin{cases} V(q) & \text{if } p \neq q \\ a & \text{if } p = q \end{cases} .$$

Given a model $\mathscr{M} = (\mathscr{F}, V)$ and a p-equivalent valuation $V\left(\begin{smallmatrix}p\\a\end{smallmatrix}\right)$ we construct a *p-equivalent* model $\mathscr{M}\left(\begin{smallmatrix}p\\a\end{smallmatrix}\right)$ with the same underlying Kripke frame $\mathscr{F}$ but with the valuation $V\left(\begin{smallmatrix}p\\a\end{smallmatrix}\right)$.

For any $w \in W$ we define:

$$\begin{aligned} \mathscr{M}, w \models p &\iff w \in V(p) \text{ if } p \text{ is a propositional variable,} \\ \mathscr{M}, w \models \neg\varphi &\iff \mathscr{M}, w \not\models \varphi\ , \\ \mathscr{M}, w \models (\varphi \wedge \psi) &\iff \mathscr{M}, w \models \varphi \text{ and } \mathscr{M}, w \models \psi\ , \\ \mathscr{M}, w \models \Diamond\varphi &\iff (w, v) \in \widehat{R} \text{ and } \mathscr{M}, v \models \varphi \text{ for some } v \in W, \end{aligned}$$

where φ, ψ are formulas of $\mathscr{L}(\Diamond)$

We say that a modal formula φ is *satisfiable* if there is some model $\mathscr{M} = (W, \widehat{R}, V)$ and some $w \in W$ such that $\mathscr{M}, w \models \varphi$, and that φ is *globally satisfiable* if there is some $\mathscr{M}$ such that $\mathscr{M}, w \models \varphi$ for each $w \in W$. We write $\mathscr{M} \models \varphi$ if φ is globally satisfiable in $\mathscr{M}$. We say that φ is *valid* on a frame $\mathscr{F}$ if $\mathscr{M} \models \varphi$ for all models $\mathscr{M}$ with underlying frame $\mathscr{F}$, and write $\mathscr{F} \models \varphi$. Furthermore, we say that φ is valid on a class of frames Λ if $\mathscr{F} \models \varphi$ for all $\mathscr{F} \in \Lambda$.

We define the notion of monotonicity of a modal formula in the same manner as we did for second-order formulas. That is to say, a formula φ is *(upwards-)monotone* with respect to some propositional variable p if for all p-equivalent models $\mathscr{M}_1 = (W, \widehat{R}, V_1)$ and $\mathscr{M}_2 = (W, \widehat{R}, V_2)$ such that $V_1(p) \subseteq V_2(p)$, we have that $\mathscr{N}_1 \models \varphi \Rightarrow \mathscr{N}_2 \models \varphi$.

We say that φ is *downwards-monotone* with respect to p if for all p-equivalent models $\mathscr{M}_1 = (W, \widehat{R}, V_1)$ and $\mathscr{M}_2 = (W, \widehat{R}, V_2)$ such that $V_1(p) \subseteq V_2(p)$, we have that $\mathscr{N}_1 \models \varphi \Leftarrow \mathscr{N}_2 \models \varphi$.

Kripke models are intuitively depicted by labelled digraphs. For example the labelled digraph in Figure 3.1 is a graphical interpretation of the model $\mathscr{M}^* = (W, \widehat{R}, V)$ where

$$\begin{aligned} W &= \{a, b, c, d, e\} \ , \\ \widehat{R} &= \{(a, a), (a, b), (b, c), (c, e), (d, b), (e, c), (e, d)\} \ , \\ V(p) &= \{a, c\} \ , \\ V(q) &= \{c, e\} \ . \end{aligned}$$

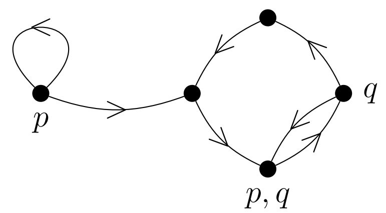


Figure 3.1: Labelled digraph depicting the model $\mathscr{M}^*$

For every $(v, w) \in \widehat{R}$, we say that w is the $\widehat{R}$-*successor* of v and that v is the $\widehat{R}$-*predecessor* of w.

In the basic modal language we can interpret $\Box\varphi$ as meaning "φ is necessary" (the alethic reading), "φ is known" (the epistemic reading), or "φ is believed to be true" (the doxastic reading), however, the basic principles of Kripke semantics hold when we deal with languages with more than one modality. For example, in the language $\mathscr{L}(P, F)$ we interpret $P\varphi$ and meaning "φ was true at some point in the past" and $F\varphi$ as meaning "φ will be true at some point in the future" (the temporal reading) and if we wish to express the knowledge of multiple agents $\alpha_1, \ldots, \alpha_n$ for which we may make use of the language $\mathscr{L}(\Diamond_{\alpha_1}, \ldots, \Diamond_{\alpha_n})$ where we interpret $\Box_{\alpha_i}\varphi$ as meaning "agent α_i knowns φ".

However, after settling on our desired interpretation of our modalities, we are faced with the problem of deciding which frames make suitable models for intended meaning of $\Box\varphi$. Surely not just any frame will do since, in the epistemic reading, if I *know* p to be true then it ought to be the case that p *is* true. That is to say $\Box p \to p$ should be valid on all frames, which it certainly isn't if we don't impose any restrictions on which frames we are allowed to consider.

As it happens, we may get around this issue if we restrict ourselves to *reflexive* frames, i.e. those for which $(x, x) \in \widehat{R}$ for all $x \in W$. We have the following proposition:

Proposition 3.1.1. *The modal formula $\Box p \to p$ is valid on the class of reflexive frames. Furthermore, if $\Box p \to p$ is valid on a frame $\mathscr{F}$ then $\mathscr{F}$ is reflexive.*

Proof: *(taken from [22])* Suppose that $\mathscr{M}$ is a model based on a reflexive frame $\mathscr{F} = (W, \widehat{R})$ and let $w \in W$. If $\mathscr{M}, w \not\models \Box p$ then trivially we have that $\mathscr{M}, w \models \Box p \to p$, so suppose $\mathscr{M}, w \models \Box p$. That is to say that $\mathscr{M}, v \models p$ for all $\widehat{R}$-successors v of w. Now since $\widehat{R}$ is reflexive, w is an $\widehat{R}$-successor of itself and so $\mathscr{M}, w \models p$. Consequently, $\Box p \to p$ is globally satisfiable in $\mathscr{M}$, but since our choice of valuation was arbitrary, $\Box p \to p$ is valid on $\mathscr{F}$.

For the converse, assume for contradiction that $\Box p \to p$ is valid on $\mathscr{F} = (W, \widehat{R})$ where $\widehat{R}$ is not reflexive. Hence, there is some $w \in W$ such that $(w, w) \notin \widehat{R}$. Let $\mathscr{M}$ be the model based on $\mathscr{F}$ with valuation $V(p) = W \setminus \{w\}$. This valuation enforces that $\mathscr{M}, w \models \Box p$ and $\mathscr{M}, w, \models \neg p$; consequently $\mathscr{M}, w \models \Box p \wedge \neg p$. Since $\Box p \to p$ is valid on $\mathscr{F}$ we have that $\mathscr{M}, w \models \Box p \to p$, which results in a contradiction. Hence $\mathscr{F}$ must be a reflexive frame. ∎

If we are interested in the epistemic interpretation of basic modal logic, then common axioms which we would like to be true of knowledge are given in Table 3.2.

Name	*Axiom*	*Epistemic Interpretation*
T	$\Box p \rightarrow p$	**True Knowledge**: Everything that is known to be true is actually true.
4	$\Box p \rightarrow \Box\Box p$	**Positive introspection**: Everything that is known, is known to be known. That is to say we have perfect knowledge of what we know.
5	$\Diamond p \rightarrow \Box\Diamond p$	**Negative introspection**: Everything that is unknown, is known to be unknown. That is to say we have perfect knowledge of what we don't know.

Table 3.2: Desirable axioms for the epistemic interpretation of $\mathscr{L}(\Diamond)$.

In a similar manner to the proof of Proposition 3.1.1, we can show that axioms **4** and **5** correspond to the first-order properties stipulating that the accessibility relation is transitive and euclidean, respectively. The logic satisfying axioms $\{\mathbf{T}, \mathbf{4}, \mathbf{5}\}$ is called **S5** and we describe modalities whose accessibility relation is reflexive, transitive and euclidean as being **S5** modalities.

This relationship between modal formulas and the corresponding class of frames on which they are valid motivates our attention toward correspondence theory, and in particular, procedures for effectively computing such correspondence properties.

3.2 Correspondence Theory

With the semantics as outlined above, there is a natural translation into second-order logic, which we call the standard translation (denoted π). For a modal formula of $\mathscr{L}(\Diamond)$, this can be effectively computed using the following rules:

$$\begin{aligned}
\pi(p, x) &= \mathbf{P}(x) \ , \\
\pi(\neg\varphi, x) &= \neg\pi(\varphi, x) \ , \\
\pi(\varphi_1 \wedge \varphi_2, x) &= \pi(\varphi_1, x) \wedge \pi(\varphi_2, x) \ , \\
\pi(\Diamond\varphi, x) &= \exists y \Big(Rxy \wedge \pi(\varphi, y) \Big) \ ,
\end{aligned}$$

where p is a propositional variable, $\mathbf{P}$ is a unique monadic second-order variable assigned to p, and R is the unique binary second-order constant associated with $\widehat{R}$.

We have, then, that for a given Kripke model $\mathscr{M} = (W, \widehat{R}, V)$ and any formula φ of $\mathscr{L}(\Diamond)$,

$$\mathscr{M}, w \models \varphi \Longleftrightarrow \mathscr{N} \models \pi(\varphi, x) \ , \tag{3.1}$$

where $\mathscr{N} = (W', *, V')$ is an $\mathscr{L}_2(R)$-model whose underlying structure W' is W, $*(R) = \widehat{R}$ and V' is any valuation such that $V'(x) = w$.

Correspondence (3.1), thus, gives us a translation of *local* satisfiability of a formula into second-order logic. We now consider global satisfiability and frame validity.

We said that a modal formula φ was globally satisfiable if $\mathscr{M}, w \models \varphi$ for all $w \in W$. Given our translation, as outlined above, this is equivalent to saying that $\mathscr{N} \models \pi(\varphi, x)$ for all models with the above underlying structure but with $V(x)$ ranging over the whole of W; that is to say that $\mathscr{N}\left(\begin{smallmatrix} x \\ w \end{smallmatrix}\right) \models \pi(\varphi, x)$ for all $w \in W$.

This is precisely the semantic definition of universal quantification in second-order logic. Hence we conclude that

$$\mathscr{M} \models \varphi \Longleftrightarrow \mathscr{N} \models \forall x \ \pi(\varphi, x) \ . \tag{3.2}$$

We now extend our correspondence theory to frame validity. We said that a modal formula φ is valid on a frame $\mathscr{F}$ if $\mathscr{M} \models \varphi$ for all models $\mathscr{M}$ with underlying frame $\mathscr{F}$.

Using correspondence (3.2), this is to say that $\mathscr{N} \models \forall x\ \pi(\varphi, x)$ for all second-order models $\mathscr{N}$, with underlying structure $(W, *)$ where $*(R) = \widehat{R}$ but with V now free to range over all possible valuations. Put differently, $\mathscr{N}\left(\begin{smallmatrix}\vec{\mathbf{P}}\\ \vec{A}\end{smallmatrix}\right) \models \forall x\ \pi(\varphi, x)$ for all $\vec{P}$-equivalent models with each $\mathbf{P}_i$ of $\vec{\mathbf{P}}$ taking the value A_i of $\vec{A}$.

This is, again, precisely the definition of universal quantification, only this time over all second-order variables occurring in $\pi(\varphi, x)$. Hence we have

$$\mathscr{F} \models \varphi \Longleftrightarrow \mathscr{N} \models \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \forall x\ \pi(\varphi, x)\ . \tag{3.3}$$

With correspondence (3.3), we see that the validity of modal axioms corresponds to some second-order property involving second-order variables. For example:

$$\mathscr{F} \models \Box p \to p \Longleftrightarrow \mathscr{N} \models \forall \mathbf{P} \forall x \Big(\forall y\, (Rxy \to \mathbf{P}y) \to \mathbf{P}x \Big)\ .$$

However, as we saw above, the axiom $\Box p \to p$ corresponded to the frame relation being reflexive; a property which is *first*-order definable involving only second-order constant symbols. Somehow we have been able to make do without the second-order variables that are introduced by the standard translation. As surprising as this is, it is a phenomenon not restricted to a few strange cases, and many of the ‘common’ properties we may wish to express about the accessibility relation $\widehat{R}$ correspond to relatively simple modal formulas, a selection of which are shown below.

Modal Axiom	*Correspondence Property*	
$\Box p \to p$	$\forall x Rxx$	(Reflexivity)
$\Box p \to \Box\Box p$	$\forall x \forall y \forall z\,(Rxy \wedge Ryz \to Ryz)$	(Transitivity)
$p \to \Box\Diamond p$	$\forall x \forall y\,(Rxy \to Ryx)$	(Symmetry)
$\Box p \to \Diamond p$	$\forall x \forall x \exists y Rxy$	(Seriality)
$\Diamond p \to \Box\Diamond p$	$\forall x \forall y \forall z\,(Rxy \wedge Rxz \to Ryz)$	(Euclideanness)

Table 3.3: Correspondence properties in $\mathscr{L}(\Diamond)$.

We might well ask then whether we can find a modal formula which defines any arbitrary first-order property of the accessibility relation? The answer, it turns out, is no, as the following result (taken from [5], page 64 –67) will demonstrate.

Definition 3.2.1. Given two models $\mathscr{M}_1 = (W_1, \widehat{R}_1, V_1)$ and $\mathscr{M}_2 = (W_2, \widehat{R}_2, V_2)$, a *bisimulation* is a non-empty binary relation $Z \subseteq W_1 \times W_2$ such that

(i) if $(w, w') \in Z$ then w and w' satisfy the same propositional variables,

(ii) if $(w, w') \in Z$, $v \in W_1$ and $(w, v) \in \widehat{R}_1$ then there is some $v' \in W_2$ such that $(v, v') \in Z$ and $(w', v') \in \widehat{R}_2$,

(iii) if $(w, w') \in Z$, $v' \in W_2$ and $(w', v') \in \widehat{R}_2$ then there is some $v \in W_1$ such that $(v, v') \in Z$ and $(w, v) \in \widehat{R}_1$,

We call models $\mathscr{M}_1$ and $\mathscr{M}_2$ *bisimilar* if there is a bisimulation between them.

Lemma 3.2.1. *Modal satisfiability in $\mathscr{M}(\Diamond)$ is invariant under bisimulations. That is to say that if Z is a bisimulation between $\mathscr{M}_1$ and $\mathscr{M}_2$ then for all $(w, w') \in Z$,*

$$\mathscr{M}_1, w \models \varphi \iff \mathscr{M}_2, w' \models \varphi$$

for all $\mathscr{L}(\Diamond)$-formulas, φ.

Proof: The proof is by induction on the length of a formula φ.

Suppose that φ is of length 1 then φ is either a propositional variable in which case the result follows from the definition or $\varphi = \perp$ in which case the result is trivial.

So, suppose that the result holds for all formulas of length less than k and suppose that φ is a formula of length k. We have three cases:

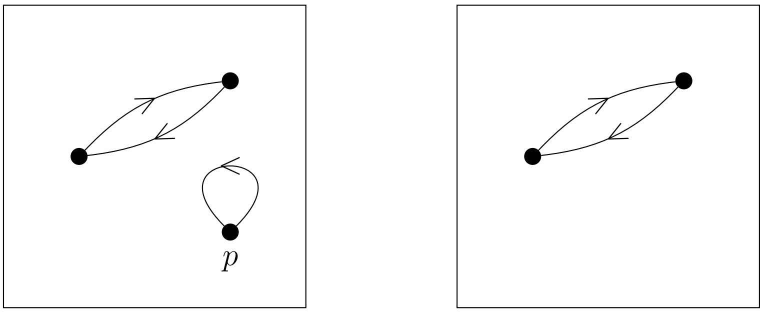
(1) If φ is of the form $\neg\psi$ for some formula ψ then $\mathscr{M}_1, w \models \neg\psi$ if and only if $\mathscr{M}_1, w \not\models \psi$. Which, by the induction hypothesis, is the case if and only if $\mathscr{M}_2, w' \not\models \psi$; which by definition, is to say that $\mathscr{M}_2, w' \models \neg\psi$.

(2) If φ is of the form $\psi_1 \wedge \psi_2$ then $\mathscr{M}_1, w \models \psi_1 \wedge \psi_2$ if and only if $\mathscr{M}_1, w \models \psi_1$ and $\mathscr{M}_1, w \models \psi_2$. Which, by the two separate applications of the induction hypothesis, is the case if and only if $\mathscr{M}_2, w' \models \psi_1$ and $\mathscr{M}_2, w' \models \psi_2$; which by definition, is to say that $\mathscr{M}_2, w' \models \psi_1 \wedge \psi_2$.

(3) The remaining case is less straightforward. Suppose φ is of the form $\Diamond\psi$ then $\mathscr{M}_1, w \models \Diamond\psi$ if and only if $\mathscr{M}_1, v \models \psi$ for some $v \in W_1$ such that $(w, v) \in \widehat{R}_1$. Now, by definition of bisimilar models, there is some $v' \in W_2$ such that $(w', v') \in \widehat{R}_2$ and $(v, v') \in Z$, hence, by the induction hypothesis, $\mathscr{M}_2, v' \models \psi$; which by definition, is to say that $\mathscr{M}_2, w' \models \Diamond\psi$. Similarly for the converse.

This completes the proof. ∎

Hence, we cannot express, by any formula of $\mathscr{L}(\Diamond)$, the first-order property $\forall x \exists y \neg Rxy$ since this formula is valid on the model $\mathscr{M}_1$ depicted below, but not on the bisimilar model $\mathscr{M}_2$.



$\mathscr{M}_1$ $\mathscr{M}_2$

In this example, we say that $\mathscr{M}_1$ is a *generated submodel* of $\mathscr{M}_2$.

Definition 3.2.2. Given a model $\mathscr{M} = (W, \widehat{R}, V)$ and a world $w \in W$, we construct the *submodel of $\mathscr{M}$ generated by w*, $\mathscr{M}' = (W', \widehat{R}', V')$ as follows

(i) Let W' be the smallest set containing w such that if $v \in W'$ and $(v, u) \in \widehat{R}$ then $u \in W'$.

(ii) Let $\widehat{R}'$ be the restriction of $\widehat{R}$ to W'; that is to say $\widehat{R}' = \widehat{R} \cap (W' \times W')$.

(iii) Let V' be the restriction of V to W'; that is to say $V' = V \cap W'$.

We have that generated submodels are a special case of bisimulations.

Proposition 3.2.1. *If $\mathscr{M}'$ is a generated submodel of $\mathscr{M}$ then $\mathscr{M}$ and $\mathscr{M}'$ are bisimilar.*

Proof: Let $\mathscr{M} = (W, \widehat{R}, V)$ and $\mathscr{M}' = (W', \widehat{R}', V')$ be a generated submodel of $\mathscr{M}$. We take Z to be the bisimulation $\{(w, w) \mid w \in W'\}$, then since $V' = V \cap W'$ we have that w satisfies the same propositional variables in both models.

Suppose that $(w, w) \in Z$, $w \in W$ and $(w, v) \in \widehat{R}$. By condition (i) of 3.2.2, v is also an element of W' and consequently, by condition (ii), $(w, v) \in \widehat{R}'$. This is condition (ii) of our definition of a bisimulation.

Finally, suppose that $(w, w) \in Z$, $w \in W'$ and $(w, v) \in \widehat{R}'$. Since $\widehat{R}'$ is a subset of $\widehat{R}$, we have that $(w, v) \in \widehat{R}$. This is the last remaining condition for Z to be a bisimulation, and so we are done. ∎

This highlights the fact that our modal operators only have access to the information held 'locally'; we cannot describe, by any modal formula, statements about worlds which are not accessible by $\widehat{R}$.

Perhaps, then, the converse might hold? Does every modal formula correspond to a first-order property of the accessibility relation? Again, the answer to this question is no, as the following theorem illustrates.

Theorem 3.2.1. *The frame validity of McKinsey's axiom $\Box\Diamond p \to \Diamond\Box p$ does not correspond to any first-order formula.*

Proof: We do not give a full proof but briefly explain the essence of the proof given in [5] (pages 133–134). They construct an uncountable frame $\mathscr{F}$ on which $\Box\Diamond p \to \Diamond\Box p$ is valid. They then show that $\Box\Diamond p \to \Diamond\Box p$ is not valid on any countable subframe of $\mathscr{F}'$ of $\mathscr{F}$.

If $\Box\Diamond p \to \Diamond\Box p$ was equivalent to some first-order property, then by the Skolem-Löwenheim downwards theorem, this property would hold on some countable subframe $\mathscr{F}'$, which in turn would correspond to the validity of $\Box\Diamond p \to \Diamond\Box p$ on $\mathscr{F}'$; a contradiction!

Hence, since $\Box\Diamond p \to \Diamond\Box p$ violates the Skolem-Löwenheim downwards theorem, it cannot have a first-order correspondent. ∎

So which formulas *do* have first-order correspondents? This innocent looking question relates yet another problem for us:

Theorem 3.2.2 (Chagrova)**.** *It is undecidable whether an arbitrary modal axiom φ has a first-order correspondence property.*

Proof: Not given. See [6]. ∎

It may then seem a bit of a fluke that we were able to find a first-order correspondence property for $\Box p \to p$, however, there are some large classes of modal formulas that *do* have first-order correspondence property which *can* be effectively computed. We will discuss the construction of these classes now.

3.3 Some Important Syntactic Classes

We first need to define the notion of *polarity* for formulas of our basic modal language $\mathscr{L}(\Diamond)$. A modal formula φ is *positive with respect to a propositional variable* p if every occurrence of p in $NNF(\varphi)$ occurs in the scope of an *even* number of negation signs and *negative w.r.t* p if every occurrence of p in $NNF(\varphi)$ occurs in the scope of an *odd* number of negation signs, where $NNF(\varphi)$ is the negation normal form of φ which can be effectively computed using the rules of Table 3.4.

A formula is *positive* (resp. *negative*) if it is positive (resp. negative) with respect to all propositional variables that occur in it.

input formula	*equivalent formula*
$\neg(\varphi \leftrightarrow \psi)$	$(\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi)$
$\varphi \leftrightarrow \psi$	$(\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$
$\neg(\varphi \rightarrow \psi)$	$\varphi \wedge \neg\psi$
$\varphi \rightarrow \psi$	$\neg\varphi \vee \psi$
$\neg(\varphi \vee \psi)$	$\neg\varphi \wedge \neg\psi$
$\neg(\varphi \wedge \psi)$	$\neg\varphi \vee \neg\psi$
$\neg\neg\varphi$	φ
$\neg\bot$	$\top$
$\neg\top$	$\bot$
$\neg\Box\varphi$	$\Diamond\neg\varphi$
$\neg\Diamond\varphi$	$\Box\neg\varphi$

Table 3.4: Rules for computing the negation normal form

Modal formulas enjoy the same nice property that second-order formulas enjoy; that polarity implies monotonicity (see Proposition 2.2.1). That is, if φ is positive with respect to some propositional variable p then φ is upwards-monotone with respect to p, and if φ is negative with respect to p then φ is down-monotone with respect to p. To see this, we simply take the standard translation of φ into second-order logic and observe that the definitions of polarity and monotonicity coincide.

Definition 3.3.1. Let σ be the singularity type of $\mathscr{L}(\Diamond)$, that is $\{\Diamond\}$ and let σ^* be the set of duals of σ, that is $\{\Box\}$.

We say a formula is *atomic* if it is a propositional variable.

A *boxed atom* is a formula of the form $\triangledown_1 \cdots \triangledown_m\, p$ for some string of boxes, $\triangledown_i \in \sigma^*$, and some propositional variable p.

A formula is a *Sahlqvist antecedent* of $\mathscr{L}(\Diamond)$ if it is a boxed atom, a propositional constant, a negative formula or of the from $\varphi \wedge \psi$, $\varphi \vee \psi$ or $\triangle\varphi$ for some $\triangle \in \sigma$ where φ, ψ are, themselves, Sahlqvist antecedents.

A *Sahlqvist implication* is a formula of the form $\varphi \rightarrow \psi$ where φ is a Sahlqvist antecedent and ψ is positive.

Sahlqvist formulas are formed from Sahlqvist implications by freely applying boxes $\triangle \in \sigma^*$, disjunctions and conjunctions. This definition follows [7] and differs from the more common definition, which only permits disjunctions between formulas that do not share any propositional variables.

Examples of Sahlqvist formulas include:

$$\Box p \to p \ , \quad \Box p \to \Box\Diamond p \ , \quad \Diamond\Box p \to \Box\Diamond p \ ,$$

$$\Diamond(p \wedge \Box\Box q) \wedge \Box q \to \Diamond(p \wedge q) \ .$$

It was proved by Sahlqvist [21] that every Sahlqvist formula corresponds to a first-order property. Moreover, these correspondence properties can be effectively computed. Methods for such computations shall be the content of Chapter 4.

To further expand upon this class we introduce generalized and simply generalized Sahlqvist formulas [14].

Definition 3.3.2. Suppose, temporarily, that $*$ is a new symbol of our language then a *universal boxed form for* $*$ is a formula of the form $*$, $\bigtriangledown\alpha$ or $A \to \alpha$ where $\bigtriangledown \in \sigma^*$, where A is some positive formula of $\mathscr{L}(\Diamond)$ not containing $*$ and α is, itself, some universal boxed form for $*$.

Now for any propositional variable p of $\mathscr{L}(\Diamond)$, a *universal boxed form for* p is a formula of the form $\chi\left\{{}^*/_p\right\}$, where χ is a universal boxed form for $*$ and $\chi\left\{{}^*/_p\right\}$ is the result of substituting all occurences of the symbol $*$ for the variable p. We say that p occurs as the *head* of $\chi\left\{{}^*/_p\right\}$ and any propositional variables occurring in χ occur in the *tail* of $\chi\left\{{}^*/_p\right\}$.

A formula is a *monadic regular formula* of $\mathscr{L}(\Diamond)$ if it is a negated universal boxed form, a propositional constant, a positive formula or of the from $\varphi \wedge \psi$, $\varphi \vee \psi$ or $\bigtriangledown\varphi$ for some $\bigtriangledown \in \sigma^*$ where φ, ψ are, themselves, monadic regular formulas. A formula is a *simply generalized Sahlqvist implication* if it is of the form $\neg\alpha \to \psi$ for some monadic regular formula α and some positive formula ψ. The class of *simply generalized Sahlqvist formulas* is built up from simply generalized Sahlqvist implications by freely applying conjunctions, disjunctions and boxes $\bigtriangledown \in \sigma^*$.

Examples of simple generalized Sahlqvist formulas that are not Sahlqvist formulas include:

$$\Box(\Box p \to p) \to \Box p \ , \qquad \Box\Big[\Box(p \to \Box q) \to (p \vee q)\Big] \vee (\Box p \to p) \ ,$$

$$\Diamond\Big(\Box(q \to \Box p) \wedge \Diamond q\Big) \to \Diamond\Box p \ .$$

Given a collection of universal boxed forms, UNI, we define a *dependency relation* $\succ_d$ on the propositional variables that occur in UNI as follows:

$$p \succ_d q \iff \begin{array}{l}\text{there is some } \chi \text{ in } UNI \text{ where } p \text{ occurs as} \\ \text{the head of } \chi \text{ and } q \text{ occurs in the tail of } \chi.\end{array}$$

We say that UNI is *acyclic* if the dependency relation on UNI is acyclic.

We now define the class of *monadic inductive formulas* as the subclass of monadic regular formulas whose component universal boxed forms form an acyclic set.

A formula is a *generalized Sahlqvist implications* if it is of the form $\neg\alpha \to \psi$ for some monadic inductive formula α and some positive formula ψ. The class of *generalized Sahlqvist formulas* is then built up from generalized Sahlqvist implications by freely applying conjunctions, disjunctions and boxes $\bigtriangledown \in \sigma^*$.

Of the examples given above for simple generalized Sahlqvist formulas, only the first $\Box(\Box p \to p) \to \Box p$, known as Löb's axiom, is not a generalized Sahlqvist formula, since it is the only one whose dependency relation, on the constituent universal boxed forms, is cyclic.

Due to their syntactic form, we have the following useful result of Conradie [7]:

Proposition 3.3.1. *Every Sahlqvist formula is semantically equivalent to a negated Sahlqvist antecedent.*

Proof: We show, by induction on the construction of φ, that if φ is a Sahlqvist formula then $NNF(\neg\varphi)$ is a negated Sahlqvist antecedent. Suppose that φ is a Sahlqvist implication of the form $\alpha \to \psi$ for some positive formula ψ, then $NNF(\neg\varphi) = \alpha \wedge NNF(\neg\psi)$, where $NNF(\neg\psi)$ is a negative formula. This is certainly in the class of Sahlqvist antecedents since α is a Sahlqvist antecedent. Suppose that φ is of the form $\Box\psi$ where ψ is a Sahlqvist formula, then $NNF(\neg\varphi) = \Diamond NNF(\neg\psi)$ where $NNF(\neg\psi)$ is, by induction, a Sahlqvist antecedent. By definition, then, $NNF(\neg\varphi)$ is a Sahlqvist antecedent. If φ is of the form $\psi_1 \wedge \psi_2$ then for Sahlqvist formulas ψ_1 and ψ_2 then $NNF(\neg\varphi) = NNF(\neg\psi_1) \vee NNF(\neg\psi_2)$. It follows then, by induction that $NNF(\neg\varphi)$ is a Sahlqvist antecedent. The remaining case where φ is of the from $\psi_1 \vee \psi_2$ is analogous. This completes the proof. ∎

We may extend this result to the class of generalized Sahlqvist formulas in an analogous way.

Proposition 3.3.2. *Every simply generalized Sahlqvist formula is semantically equivalent to a monadic regular formula and every generalized Sahlqvist formula is semantically equivalent to a monadic inductive formula.*

Proof: The proof that if φ is a simply generalized Sahlqvist formula then $NNF(\varphi)$ is a monadic regular formula is analogous to the previous result. The base case is the only noteworthy alteration: if φ is a generalized Sahlqvist implication of the form $\neg\alpha \to \psi$, for some monadic regular formula α and some positive formula ψ, then $NNF(\varphi) = \alpha \vee \psi$. Now, ψ is a positive formula, and, since the class of monadic regular formulas subsumes the class of positive formulas and is closed under disjunction, we must have that $NNF(\varphi)$ is a monadic regular formula. The induction step is identical to those given in the proof of Proposition 3.3.1.

For the second part of this proposition, we note that since the component universal boxed forms are unaffected by this transformation to negation normal form, it follows that if φ is a generalized Sahlqvist formula then $NNF(\varphi)$ is a monadic inductive formula. ∎

A trivial point to note then is that all negated Sahlqvist formulas $\neg\varphi$ are semantically equivalent to Sahlqvist implications of the form $\varphi \rightarrow \perp$ and, similarly, all monadic regular (resp. inductive) formulas $\neg\psi$ are semantically equivalent to simply generalized Sahlqvist implications (resp. generalized Sahlqvist implications) of the from $\psi \rightarrow \perp$.

It can be shown [14] that all generalized Sahlqvist formulas, like Sahlqvist formulas, have first-order correspondence properties that can be effectively computed. We will detail some algorithms for effectively computing these correspondence properties in Chapter 4.

We will show in Chapter 4 that the successful computation of correspondence properties for these classes is, in part, due to the syntactic form of boxed atoms and universal boxed forms. Both have standard translations which are semantically equivalent to *PIA-conditions.*

Definition 3.3.3. A first-order formula is a *'Positive Implies Atom'-condition* (or *PIA-condition* for short) [25] if it is of the form

$$\forall y\,[\beta(\mathbf{P}, y) \rightarrow \mathbf{P}y]\ ,$$

where $\beta(\mathbf{P}, y)$ is some positive formula with free variables $\mathbf{P}$ and y and $\mathbf{P}y$ is the standard translation of an atomic modal formula.

We note that atomic formulas, boxed atoms and universal boxed forms all have standard translations which are semantically equivalent to PIA-conditions; a fact we shall exploit in the following chapter, where we focus on how exactly we go about computing first-order correspondence properties.

3.4 The Modal Logic $\mathscr{L}(\Diamond, D)$

As a consequence of Proposition 3.2.1, we have seen that the modalities $\Diamond$ and $\Box$ only operate locally and, thus, can only make statements about all those worlds w which are locally accessible by $\widehat{R}$. We now introduce a modal operator of a very different flavour. The language $\mathscr{L}(\Diamond, D)$ is an extension of the basic modal language by adding a new unary existential modality D; the *difference operator*.

Definition 3.4.1. The modal formulas of the language $\mathscr{L}(\Diamond, D)$ are formed as follows:

$$\varphi ::= p \ \mid \ \neg\varphi \ \mid \ (\varphi_1 \wedge \varphi_2) \ \mid \ \Diamond\varphi \ \mid \ D\varphi$$

where p is a propositional variable.

We use the abbreviations of Table 3.1 together with the addition of abbreviation given in Table 3.5.

name	*abbreviation*	*equivalent formula*
dual difference operator	$\overline{D}\varphi$	$\neg D \neg\varphi$

Table 3.5: Additional abbreviation for $\mathscr{L}(\Diamond, D)$.

The semantics of $\mathscr{L}(\Diamond, D)$ are as before but with the additional definition that

$$\mathscr{M}, x \models D\varphi \iff \text{there is some } y \neq x \text{ such that } \mathscr{M}, y \models \varphi \ ,$$

which leads to a natural extension of the standard translation by the rule:

$$\pi(D\varphi, x) = \exists y \Big(y \not\approx x \wedge \pi(\varphi, y) \Big) \ .$$

The difference operator is of interest because it allows far more expressive power into our language. That is to say, we can formulate statements that are not semantically equivalent to any formula of the basic modal language. For example, $\mathscr{L}(\Diamond, D)$ is able to distinguish between the bisimilar models $\mathscr{M}_1$ and $\mathscr{M}_2$, depicted earlier and repeated below in Figure 3.2, with the formula $p \vee Dp$.

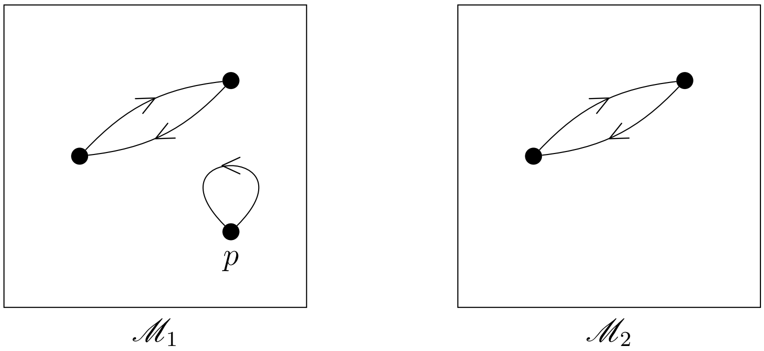


Figure 3.2: Bisimilar models that can be disginguished by the $\mathscr{L}(\Diamond, D)$-formula $(p \vee Dp)$

That is to say, there is some formula φ of $\mathscr{L}(\Diamond, D)$ and some element w in the universe of $\mathscr{M}_1$ such that $\mathscr{M}_1, w \models \varphi$ but $\mathscr{M}_2 \not\models \varphi$.

Some first-order correspondence properties expressible in $\mathscr{L}(\Diamond, D)$ that are not expressible in $\mathscr{L}(\Diamond)$ are given in Table 3.6.

Modal Axiom	*Correspondence Property*	
$\big(p \wedge \overline{D}\neg p\big) \rightarrow \Box\neg p$	$\forall x \neg Rxx$	(Irreflexivity)
$\big(p \wedge \overline{D}\neg p\big) \rightarrow \Box\Box\neg p$	$\forall x \forall y\, (Rxy \rightarrow \neg Ryx)$	(Asymmetry)
$\Big(p \wedge \Diamond(\Diamond p \wedge \neg p)\Big) \rightarrow \overline{D}p$	$\forall x \forall y\, (Rxy \wedge Ryx \rightarrow x \approx y)$	(Anti-symmetry)
$p \vee Dp \rightarrow \Diamond p$	$\forall x \forall y \;\; Rxy$	$(\widehat{R} = W^2)$

Table 3.6: Correspondence properties in $\mathscr{L}(\Diamond, D)$, adapted from [8].

We can extend the class of Sahlqvist formulas and generalized Sahlqvist formulas given in Definition 3.3.1 to the language $\mathscr{L}(\Diamond, D)$ by letting σ be the similarity type $\{\Diamond, D\}$ and σ^* be the set of duals, $\{\Box, \overline{D}\}$.

We have that all Sahlqvist formulas of $\mathscr{L}(\Diamond, D)$ have first-order correspondence properties which can be effectively computed [8]. In fact we will show that every generalized Sahlqvist formula of this language has a first-order correspondence property and that every simply generalized Sahlqvist formula of $\mathscr{L}(\Diamond, D)$ has a correspondence property in the first-order logic of fixed-points.

3.5 Global and Graded Modalities

Some other languages of interest are $\mathscr{L}(\Diamond, \langle\exists\rangle)$ and $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$ where $\langle\exists\rangle$ is 'the global existential modality' and $(\geq n)$ is the 'greater than or equal modality'. Formulas of these languages are formed in the usual way, where $\langle\exists\rangle$ and $(\geq n)$ are unary modalities. We also make use of the abbreviations of Table 3.7.

name	*abbreviation*	*equivalent formula*
global universal modality	$\boxed{\forall}\varphi$	$\neg\langle\exists\rangle\neg\varphi$
less than or equal modality	$(\leq n)\varphi$	$(\geq n+1)\varphi$
exactly modality	$(= n)\varphi$	$\big((\leq n)\varphi \wedge (\geq n)\varphi\big)$

Table 3.7: Abbreviations for languages that include global and graded modalities.

We extend the semantics of the basic modal language to formulas of the form $\langle\exists\rangle\varphi$ and $(\geq n)\varphi$ as follows:

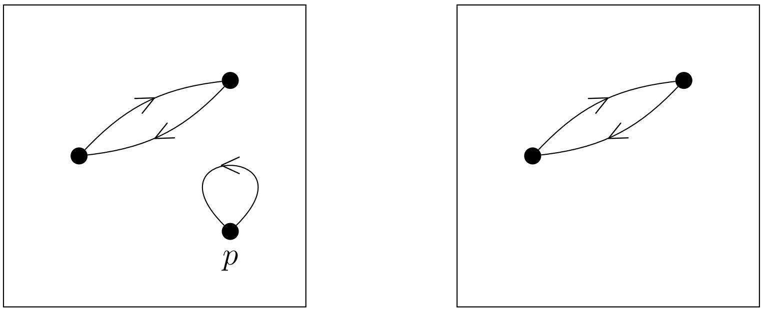
$$\begin{aligned} \mathscr{M}, x \models \langle\exists\rangle\varphi \quad &\Longleftrightarrow \quad \text{there is some } y \in W \text{ such that } \mathscr{M}, y \models \varphi \ , \\ \mathscr{M}, x \models (\geq n)\varphi \quad &\Longleftrightarrow \quad \text{card}\,\{y \in W \mid \ \text{such that } (x,y) \in R \text{ and } \mathscr{M}, y \models \varphi\} \geq n \ . \end{aligned}$$

where $\text{card}(X)$ is the cardinality of the set X.

Again, we have that the expressive power of both these languages strictly extend that of the basic modal language $\mathscr{L}(\Diamond)$.

Proposition 3.5.1. *The expressive power of the language $\mathscr{L}(\Diamond, \langle\exists\rangle)$ strictly extends that of $\mathscr{L}(\Diamond)$.*

Proof: Clearly every formula of $\mathscr{L}(\Diamond)$ is also a formula of $\mathscr{L}(\Diamond, \langle\exists\rangle)$, however, since modal satisfiability in $\mathscr{L}(\Diamond)$ is invariant under bisimulation, $\mathscr{L}(\Diamond)$ cannot distinguish between the bisimilar models $\mathscr{M}_1$ and $\mathscr{M}_2$ depicted below.



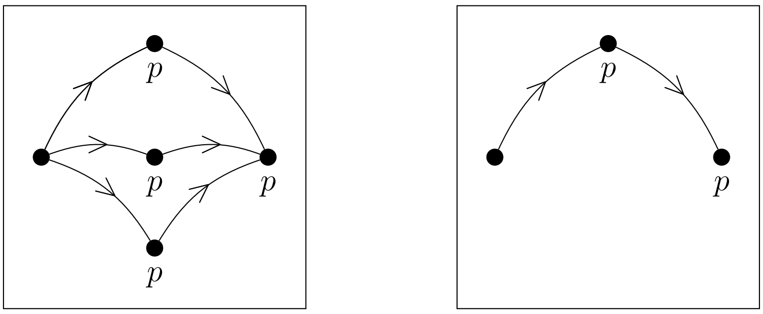
$\mathscr{M}_1$ $\mathscr{M}_2$

However, models $\mathscr{M}_1$ and $\mathscr{M}_2$ can be distinguished by the $\mathscr{L}(\Diamond, \Diamond\!\!\!\!\exists)$-formula $\Diamond\!\!\!\!\exists p$. Hence, $\mathscr{L}(\Diamond, \Diamond\!\!\!\!\exists)$ is strictly more expressive than $\mathscr{L}(\Diamond)$. ∎

Proposition 3.5.2. *The expressive power of the language $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$ strictly extends that of $\mathscr{L}(\Diamond)$.*

Proof: First, we note that the definition of $\Diamond\varphi$ is precisely that of $(\geq 1)\varphi$ so that $\mathscr{L}(\Diamond)$ is surely a definable fragment of $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$.

However, since modal satisfiability in $\mathscr{L}(\Diamond)$ is invariant under bisimulations, $\mathscr{L}(\Diamond)$ cannot distinguish between the bisimilar models $\mathscr{M}_1$ and $\mathscr{M}_2$ depicted below.



$\mathscr{M}_1$ $\mathscr{M}_2$

However, these models can be distinguished by the $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$-formula $(p \vee (\geq 3)p)$. Hence, $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$ is also strictly more expressive than $\mathscr{L}(\Diamond)$. ∎

As with the language $\mathscr{L}(\Diamond, D)$, we can extend the standard translation to formulas of the form $\langle\exists\rangle\varphi$ and $(\geq n)\varphi$ with the following rules:

$$
\begin{aligned}
\pi(\langle\exists\rangle\varphi, x) &= \exists y \ \pi(\varphi, y) \ , \\
\pi((\geq n)\varphi, x) &= \exists y_1 \ldots \exists y_n \left[Rxy_1 \wedge \ldots \wedge Rxy_n \wedge \bigwedge_{i \neq j} y_i \not\approx y_j \wedge \bigwedge_{i=1}^{n} \pi(\varphi, y_i) \right] \ .
\end{aligned}
$$

The concept of a global modality seems like a useful addition to our modal language which allows for even more expressive power over $\mathscr{L}(\Diamond, D)$. However, this is not the case as $\langle\exists\rangle$ is already definable in $\mathscr{L}(\Diamond, D)$ by the formula $\langle\exists\rangle\varphi := \varphi \vee D\varphi$. As so, we may use $\langle\exists\rangle$ and $[\forall]$ as additional abbreviations in the language $\mathscr{L}(\Diamond, D)$, and will do so in later chapters.

We note, however, that $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$ is strictly more expressive than $\mathscr{L}(\Diamond, D)$. To show this we need to slightly weaken the definition of a bisimulation.

Definition 3.5.1. Given two models $\mathscr{M}_1 = (W_1, \widehat{R}_1, V_1)$ and $\mathscr{M}_2 = (W_2, \widehat{R}_2, V_2)$, a $\neq$*-bisimulation* [9] is a non-empty binary relation $Z \subseteq W_1 \times W_2$ such that

(i) if $(w, w') \in Z$ then w and w' satisfy the same propositional variables,

(ii) if $(w, w') \in Z$, $v \in W_1$ and $(w, v) \in \widehat{R}_1$ then there is some $v' \in W_2$ such that $(v, v') \in Z$ and $(w', v') \in \widehat{R}_2$,

(iii) if $(w, w') \in Z$, $v' \in W_2$ and $(w', v') \in \widehat{R}_2$ then there is some $v \in W_1$ such that $(v, v') \in Z$ and $(w, v) \in \widehat{R}_1$,

(iv) if $(w, w') \in Z$, $v \in W_1$ and $w \neq v$ then there is some $v' \in W_2$ such that $(v, v') \in Z$ and $w' \neq v'$,

(v) if $(w, w') \in Z$, $v' \in W_2$ and $w' \neq v'$ then there is some $v \in W_1$ such that $(v, v') \in Z$ and $w \neq v$.

We call models $\mathscr{M}_1$ and $\mathscr{M}_2$ $\neq$*-bisimilar* if there is a $\neq$-bisimimulation between them.

We have, then, the following lemma.

Lemma 3.5.1. *Modal satisfiability in $\mathscr{L}(\Diamond, D)$ is invariant under $\neq$-bisimulations.*

Proof: The proof is by induction on the length of a formula φ. The cases where φ is of the form $\neg\psi$, $\psi_1 \wedge \psi_2$, and $\Diamond\psi$ are dealt with as in the proof of lemma 3.2.1, the only case remaining is where φ is of the form $D\psi$, which we prove here.

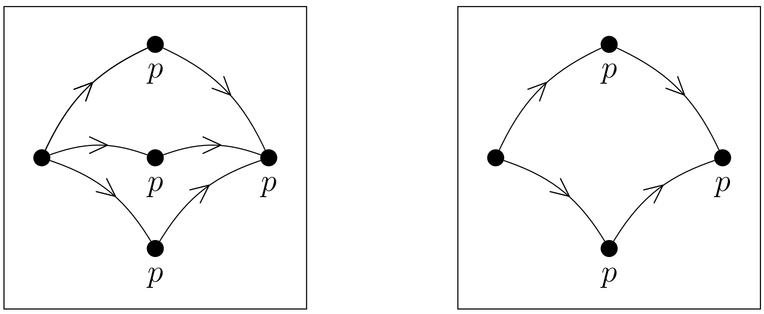
Suppose $\mathscr{M}, w \models D\psi$ then by definition there is some $v \neq w$ such that $\mathscr{M}, v \models \psi$. By the induction hypothesis we have that $\mathscr{M}', v' \models \psi$. The side conditions ensure we can choose v such that $v' \neq w'$ from which we conclude that $\mathscr{M}', w' \models D\psi$.

This completes the remaining induction case. ∎

From this, Proposition 3.5.3 follows.

Proposition 3.5.3. *The language $\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$ is strictly more expressive than $\mathscr{L}(\Diamond, D)$.*

Proof: The language of graded modalities can distinguish between the $\neq$-bisimilar models $\mathscr{M}_1$ and $\mathscr{M}_2$, depicted below, with the formula $(p \vee (\geq 3)p)$, whereas $\mathscr{L}(\Diamond, D)$ cannot.



$\mathscr{M}_1$ $\mathscr{M}_2$

∎

We will return to graded modalities again in Chapter 7. The last modal language we need to introduce is $\mathscr{L}_n^{\smile}(\Diamond)$.

3.6 The Language $\mathscr{L}_n^{\smile}(\Diamond)$

This extension of the basic modal language will be the target language of the MA-calculus, which we introduce in the Chapter 4 for computing first-order correspondence properties for formulas of $\mathscr{L}(\Diamond)$.

Definition 3.6.1. The *hybrid modal language* $\mathscr{L}_n^{\smile}(\Diamond)$ is an extension of $\mathscr{L}(\Diamond)$ by the symbols $\Diamond^{\smile}$ ('converse possibility') and a countable set of nominal symbols, usually denoted by lower case Roman characters.

The formulas of $\mathscr{L}_n^{\smile}(\Diamond)$ are given by the rule

$$\varphi ::= p \mid a \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \Diamond\varphi \mid \Diamond^{\smile}\varphi\ ,$$

where p is a propositional variable and a is a nominal symbol.

We make use of the usual conventions for omitting brackets and the additional abbreviation given in Table 3.8.

name	*abbreviation*	*equivalent formula*
converse necessity	$\Box^{\smile}\varphi$	$\neg\Diamond^{\smile}\neg\varphi$

Table 3.8: Abbreviations for $\mathscr{L}_n^{\smile}(\Diamond)$.

We define a *hybrid Kripke frame* to be a tuple $(W, \widehat{R}, *)$ where, as before, W is a non-empty set, $\widehat{R}$ is a binary relation on W and, additionally, $*$ is a function that maps every nominal symbol, a, to an element of W. A *valuation* on $\mathscr{F}$ is a function, V, which maps propositional variables to subsets of W. A *hybrid Kripke model* is a hybrid Kripke frame $\mathscr{F}$ coupled with a valuation V on $\mathscr{F}$. Truth of a hybrid modal formula φ is then defined inductively on the length of φ as follows.

For a given hybrid model $\mathscr{M} = (W, \widehat{R}, *, V)$ and any $w \in W$,

$$\begin{aligned}
\mathscr{M}, w \models p &\iff w \in V(p) \text{ if } p \text{ is a propositional variable,}\\
\mathscr{M}, w \models a &\iff w = *(a) \text{ if } a \text{ is a nominal,}\\
\mathscr{M}, w \models \neg\varphi &\iff \mathscr{M}, w \not\models \varphi \ ,\\
\mathscr{M}, w \models \neg\varphi_1 \wedge \varphi_2 &\iff \mathscr{M}, w \models \varphi_1 \text{ and } \mathscr{M}, w \models \varphi_2 \ ,\\
\mathscr{M}, w \models \Diamond\varphi &\iff (w, v) \in \widehat{R} \text{ and } \mathscr{M}, v \models \varphi \text{ for some } v \in W,\\
\mathscr{M}, w \models \Diamond^{\smile}\varphi &\iff (v, w) \in \widehat{R} \text{ and } \mathscr{M}, v \models \varphi \text{ for some } v \in W.
\end{aligned}$$

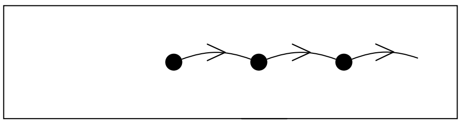
Nominals act like constants and allow us to 'name' individual worlds of our universe with formulas of $\mathscr{L}_n^{\smile}(\Diamond)$, whereas converse modalities allow us to 'look' backwards along the binary relation of $\Diamond$.

We can extend the standard translation to formulas of this language by providing the rules

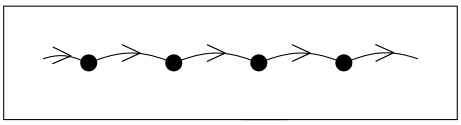
$$\begin{aligned}
\pi(a, x) &= \bar{a} \approx x \ ,\\
\pi(\Diamond^{\smile}\varphi, x) &= \exists y\Big(Ryx \wedge \pi(\varphi, y)\Big) \ ,
\end{aligned}$$

where $\bar{a}$ is a unique constant symbol assigned to the nominal a and R is the same second-order constant symbol assigned to $\widehat{R}$.

We note that this is, indeed, a strict extension of the expressive power of the basic modal logic $\mathscr{L}(\Diamond)$, since it can distinguish between the bisimilar models $\mathscr{M}_1 = \Big(\mathbb{N},\ \{(n, n+1) \mid n \in \mathbb{N}\},\ \emptyset\Big)$ and $\mathscr{M}_2 = \Big(\mathbb{Z},\ \{(n, n+1) \mid n \in \mathbb{Z}\},\ \emptyset\Big)$, depicted below, with the formula $\Box^{\smile} \bot$, whereas $\mathscr{L}(\Diamond)$ cannot.



$\mathscr{M}_1$



$\mathscr{M}_2$

Chapter 4

Computing Correspondence Properties

The key to the methods we present for elimination of second-order variables is Ackermann's Lemma introduced in [2]. We do not prove it here but instead refer the reader to [13] (pages 75–76) for a modern proof of the lemma.

Lemma 4.0.1 (Ackermann's Lemma)**.** *Let $\boldsymbol{P}$ be an n-ary second-order variable, $\vec{y}$ be an n-tuple of first-order variables and $\vec{x}$ another tuple of first-order variables. Let α be a formula which contains no occurrence of the variable $\boldsymbol{P}$ and NEG (resp. POS) be a formula which is negative (reps. positive) with respect to $\boldsymbol{P}$. We then have the following equivalences:*

$$\exists \boldsymbol{P}\Big(\forall \vec{y}\,[\alpha(\vec{y},\vec{x}) \rightarrow \boldsymbol{P}(\vec{y})] \wedge NEG(\boldsymbol{P})\Big) \equiv NEG\left\{{}^{\boldsymbol{P}(\vec{y})}/_{\alpha(\vec{y},\vec{x})}\right\}$$

and

$$\exists \boldsymbol{P}\Big(\forall \vec{y}\,[\boldsymbol{P}(\vec{y}) \rightarrow \alpha(\vec{y},\vec{x})] \wedge POS(\boldsymbol{P})\Big) \equiv POS\left\{{}^{\boldsymbol{P}(\vec{y})}/_{\alpha(\vec{y},\vec{x})}\right\} \ .$$

We present two well known procedures for computing first-order correspondence properties of formulas of the basic modal language $\mathscr{L}(\Diamond)$, namely the Modal Ackermann calculus, underlying the algorithms MSQEL and SQEMA [13] and the method of Sahlqvist and van Benthem [21, 24]. We also present a second-order version of the Modal Ackermann calculus, SOQE, from [22], which acts on sets of $\mathscr{L}_2$-formulas.

We are concerned only with the relative computational capabilities of these algorithms, and so do not consider the issue of various optimizations that exist.

4.1 The MA-Calculus

The Modal Ackermann (MA) calculus [23] consists of a set of rewrite rules that act on formulas of $\mathscr{L}_n^{\smile}(\Diamond)$. Each of the rules preserve equivalence in the sense that, if N/M is a rule of the calculus then $\bigwedge N$ is globally satisfiable if and only if $\bigwedge M$ is globally satisfiable. That is to say that there is a hybrid model $\mathscr{N}$ such that $\mathscr{N} \models \bigwedge N$ if and only if there is a hybrid model $\mathscr{M}$ such that $\mathscr{M} \models \bigwedge M$. The calculus manipulates modal formulas until we are in a position to apply the modal form of Ackermann's Lemma.

Lemma 4.1.1 (Modal Ackermann Lemma)**.** *Let p be a propositional variable and let α be a modal formula not containing p. Let NEG be a modal formula which is negative with respect to p. We have that $\mathscr{M} \models NEG\,\{{}^{p}/_{\alpha}\}$ if and only if there is a p-equivalent model $\mathscr{M}'$ such that $\mathscr{M}' \models (\alpha \to p) \wedge NEG(p)$.*

Proof: This is immediate from Lemma 4.0.1 and the definitions given in Section 3.2. ∎

We will make use of the notation $\varphi\,\{{}^{\alpha}/_{\beta}\}$, which represents the result of uniformly substituting all occurrence of the string of symbols α in φ for the string of symbols β. For example:

$$\Big(p \wedge \Box(\Diamond p \wedge q)\Big)\,\big\{{}^{\Diamond p}/_{[\Box q \to p]}\big\} \;=\; p \wedge \Box([\Box q \to p] \wedge q)\ .$$

Compound substitutions are written as $\varphi\,\{{}^{\alpha_1}/_{\beta_1}\}\,\{{}^{\alpha_2}/_{\beta_2}\}$, which represents the result of first substituting all occurences of α_1 for β_1 and *then* substituting all occurences of α_2 for β_2. For example:

$$\Big(p \wedge \Box(\Diamond p \wedge q)\Big)\,\big\{{}^{\Diamond p}/_{[\Box q \to p]}\big\}\,\{{}^{q}/_{\neg q}\} \;=\; p \wedge \Box([\Box \neg q \to p] \wedge \neg q)\ .$$

In contrast we may use the notation $\varphi\,\big\{{}^{(\alpha_1,\ldots,\alpha_n)}/_{\beta_1,\ldots,\beta_n)}\big\}$ for the result of substituting all occurrences of α_i for β_i for all $1 \leq i \leq n$, *at the same time*, provided no α_j is a subformula of any α_k. For example:

$$\Big(p \wedge \Box(\Diamond p \wedge q)\Big)\,\big\{{}^{(\Diamond p,\ q)}/_{([\Box q \to p],\ \neg q)}\big\} \;=\; p \wedge \Box([\Box q \to p] \wedge \neg q)\ .$$

We are now equipped to describe the MA-calculus.

Given a modal formula Φ of $\mathscr{L}(\Diamond)$, we select an ordering $<$ on the propositional variables occurring in Φ and non-deterministically apply the rules of MA, given in Tables 4.1–4.2, to the set of hybrid clauses $\{\neg a \vee \neg\Phi\}$, where a is a fresh nominal. This clause set corresponds to a refutation of Φ at a in some hybrid model; that is to say, $\{\neg a \vee \neg\Phi\}$ is satisfiable if and only if Φ is not valid.

Surfacing $$\frac{N \cup \{\alpha \vee \Box\beta(p)\}}{N \cup \{\Box^{\smile}\alpha \vee \beta(p)\}}$$

side conditions:

(i) β is positive with respect to the minimal propositional variable p,

(ii) p does not occur in α.

Skolemization $$\frac{N \cup \{\neg a \vee \Diamond\beta(p)\}}{N \cup \{\neg a \vee \Diamond b, \neg b \vee \beta(p)\}}$$

where b is a new nominal, provided that p is minimal with respect to $<$ and β is negative with respect to p.

side conditions:

(i) β is positive with respect to the minimal propositional variable p.

Reduce $$\frac{N \cup \{\gamma, \alpha\}}{N \cup \{\gamma, \beta\}}$$

side conditions:

(i) $\mathscr{M} \models \gamma \wedge \alpha$ if and only if $\mathscr{M} \models \gamma \wedge \beta$ for any Kripke model $\mathscr{M}$.

Table 4.1: The MA-calculus rules (1/2)

Clausify

$$\frac{N \cup \{\neg(\alpha \vee \beta)\}}{N \cup \{\neg\alpha, \neg\beta\}}$$

side conditions: none

Ackermann

$$\frac{N \cup \{\alpha_1 \vee p, \ldots, \alpha_n \vee p\}}{N\left\{{}^{p}/_{(\neg\alpha_1 \vee \ldots \vee \neg\alpha_n)}\right\}}$$

side conditions:

(i) p is the minimal propositional variable with respect to $<$,

(ii) N is negative with respect to p,

(iii) p does not occur in any of the α_i.

Purify$^+$

$$\frac{N}{N\left\{{}^{p}/_{\top}\right\}}$$

side conditions:

(i) p is the minimal propositional variable with respect to $<$,

(ii) N is positive with respect to p.

Purify$^-$

$$\frac{N}{N\left\{{}^{p}/_{\bot}\right\}}$$

side conditions:

(i) p is the minimal propositional variable with respect to $<$,

(ii) N is negative with respect to p.

Table 4.2: The MA-calculus rules (2/2)

If the MA-calculus terminates successfully then we will have reduced $\{\neg a \vee \neg\Phi\}$ to a set of clauses, N, which contain no propositional variables. Consequently, the standard translation of $\bigwedge N$ is a first-order formula η. We have that Φ is not valid if and only if $\{\neg a \vee \neg\Phi\}$ is satisfiable, which is the case if and only if the first-order formula η is satisfiable, which is the case if and only is $\neg\eta$ is not valid. Hence, the frame validity of Φ corresponds to the first-order property $\neg\eta$.

In the event that the calculus is unable to reduce $\{\neg a \vee \neg\Phi\}$ to a variable-free set of clauses, we must select a different ordering and repeat the process again in the hope of better success. We are not guaranteed successful termination, since the validity of formulas such as McKinsey's axiom $\Box\Diamond p \rightarrow \Diamond\Box p$ have no first-order correspondent (see Proposition 3.2.1), but what is interesting is that, where one ordering may fail, another ordering may succeed so we may need to try all such orderings to be sure of the success or failure of the procedure. This is illustrated by Example 4.1.1.

Example 4.1.1. The MA-calculus fails to compute a first-order correspondence property for the frame validity of

$$\Box\left(\Box\neg p \leftrightarrow q\right) \rightarrow \Diamond\Box p \tag{4.1}$$

under the ordering $p < q$, however, succeeds under the ordering $q < p$. (Example taken from [23], where the polarity of p appears reversed.)

We non-deterministically apply the rules of Tables 4.3–4.6 to the clause set $\left\{\neg a \vee \left(\Box\left(\Box\neg p \leftrightarrow q\right) \wedge \neg\Diamond\Box p\right)\right\}$ as follows:

~~1.~~	$\neg a \vee \left(\Box\left(\Box\neg p \leftrightarrow q\right) \wedge \neg\Diamond\Box p\right)$	
~~2.~~	$\neg a \vee \Box\left(\Box\neg p \leftrightarrow q\right)$	Clausify, 1
3.	$\neg a \vee \neg\Diamond\Box p$	Clausify, 1
~~4.~~	$\Box^{\smile}\neg a \vee \left(\Box\neg p \leftrightarrow q\right)$	Surfacing, 2
~~5.~~	$\Box^{\smile}\neg a \vee \left[\left(\Diamond p \vee q\right) \wedge \left(\neg q \vee \Box\neg p\right)\right]$	Reduce, 4
6.	$\neg a \vee \Diamond p \vee q$	Clausify, 5
7.	$\neg a \vee \neg q \vee \Box\neg p$	Clausify, 5

The algorithm now terminates without eliminating p since we have no rule to bring the positive occurrence of p in clause 6 to the surface.

Suppose, however, that we had used the ordering $q < p$. Our algorithm would proceed as above to produce the set of clauses $\{3, 6, 7\}$ but, in this case, the algorithm is able to continue with its execution:

3.	$\neg a \vee \neg\Diamond\Box p$	
~~6.~~	$\neg a \vee \Diamond p \vee q$	
~~7.~~	$\neg a \vee \neg q \vee \Box\neg p$	
~~8.~~	$\neg a \vee \neg\neg(\neg a \vee \Diamond p) \vee \Box\neg p$	Ackermann 6 into 7
9.	$\neg a \vee \Diamond p \vee \Box\neg p$	Reduce, 8

Clause 9 contains both $\Diamond p$ and it's negation $\Box\neg p$, and is, therefore, semantically equivalent to $\top$.

10.	$\top$	Simplification, 9
11.	$\neg a \vee \neg\Diamond\Box\bot$	Purify^{-}, 3

The algorithm now terminates, having successfully eliminated all propositional variables. We take the standard translation of clauses 10 and 11:

$$\top \wedge \neg\exists y\Big(Ray \wedge \forall z[Ryz \rightarrow\bot]\Big)$$

We now unskolemize [18], replacing all occurences of the nominal a with the first-order variable x, which we existentially quantify, to give us

$$\exists x\Big[\top \wedge \neg\exists y\Big(Rxy \wedge \forall z[Ryz \rightarrow\bot]\Big)\Big] \ .$$

This now simplifies to give us $\exists x\neg\exists y\exists y(Ray \wedge \forall z\neg Ryz)$, which is satisfiable if and only if (4.1) is refutable. Consequently, the frame validity of (4.1) corresponds to the first-order formula $\forall x\exists y\,(Rxy \wedge \forall z\neg Ryz)$.

Although we have demonstrated that the success of the MA-calculus is largely dependent on the ordering we choose for the propositional variables, we do, however, have the following result of Schmidt [23]:

Theorem 4.1.1 (Schmidt)**.** *The MA-calculus successfully computes the first-order correspondence properties of monadic inductive formulas for* any *ordering on the propositional variables.*

Consequently, since every negation of any generalized Sahlqvist formula is a monadic inductive formula (see Proposition 3.3.2), the MA-calculus can be used to successfully compute the first-order correspondence properties for the frame validity of generalized Sahlqvist Formulas. Furthermore, any ordering on the propositional variables will lead to a successful computation.

4.2 The SOQE-Calculus

The next algorithm we present is a second-order adaptation of the MA-calculus called the SOQE-calculus (short for the Second-Order Quantifier Elimination calculus), which acts on sets of $\mathscr{L}_2$-formulas [22].

As with the MA-calculus, the fundamental idea to this procedure is to manipulate clauses until we are in a position to apply Ackermann's lemma. Each of the rules preserve equivalence in the sense that if N/M is a rule of the calculus then $\exists\vec{P}\forall\vec{x}\bigwedge N \equiv \exists\vec{P}\forall\vec{x}\bigwedge M$ where $\vec{P}$ (resp. $\vec{x}$) is the vector of all the second-order (resp. first-order) variables that occur free in $\bigwedge N \cup M$.

Given a modal formula Φ of $\mathscr{L}(\Diamond)$, we select an ordering $<$ on the propositional variables that occur in Φ and non-deterministically apply the rules of SOQE, given in Tables 4.3 – 4.6, to the set $\Big\{\exists x\ NNF\Big(\neg\pi(\Phi, x)\Big)\Big\}$ where the negation normal form of $\neg\pi(\Phi, x)$ can be effectively computed using the rules of Table 3.4.

∧-Eliminate
$$\frac{N \cup \{\alpha \vee (\beta_1 \wedge \beta_2)\}}{N \cup \{\alpha \vee \beta_1, \alpha \vee \beta_2\}}$$

side conditions:

(i) The minimal second-order variable **P** (with respect to $<$) occurs in the main premise.

∀-Eliminate
$$\frac{N \cup \{\alpha \vee \forall x \;\; \beta(x)\}}{N \cup \{\alpha \vee \beta \{x/y\}\}}$$

where y is a fresh first-order variable.
side conditions:

(i) The main premise is positive with respect to the minimal second-order variable **P**.

∃-Eliminate
$$\frac{N \cup \{\alpha \vee \exists x \;\; \beta(x)\}}{N \cup \{\alpha \vee \beta \{x/a\}\}}$$

where a is a fresh first-order constant.
side conditions:

(i) β is positive with respect to the minimal second-order variable **P**,

(ii) β does not contain any free variables.

Table 4.3: The SOQE-calculus (1/4)

Term abstraction
$$\frac{N \cup \{\alpha \vee P(a)\}}{N \cup \{\alpha \vee x \not\approx a \vee P(x)\}}$$

where x is a fresh first-order variable.
side conditions:

(i) $\mathbf{P}$ is the minimal second-order variable with respect to $<$.

Variable renaming
$$\frac{N \cup \{\alpha(x)\}}{N \cup \{\alpha\,\{x/y\}\}}$$

side conditions:

(i) α is positive with respect to the minimal second-order variable $\mathbf{P}$,

(ii) x occurs freely in α,

(iii) y does not occur in α.

Simplification
$$\frac{N \cup \{\gamma, \alpha\}}{N \cup \{\gamma, \beta\}}$$

side conditions:

(i) $\forall \vec{x}\,(\gamma \wedge \alpha) \equiv \forall \vec{x}\,(\gamma \wedge \beta)$,

(ii) the depth of β is less than the depth of α.

Table 4.4: The SOQE-calculus (2/4)

Ackermann
$$\frac{N \cup \{\alpha_1(x) \vee \mathbf{P}(x), \ldots, \alpha_n(x) \vee \mathbf{P}(x)\}}{N\left\{^{\mathbf{P}(u)}/_{(\sim\alpha_1 \vee \ldots \vee \sim\alpha_n)(u)}\right\}}$$

side conditions:

(i) the minimal second-order variable $\mathbf{P}$ does not occur in any of the α_i,

(ii) N is negative with respect to $\mathbf{P}$,

(iii) N and α_i do not share any of the same first-order variables. α_i and α_j do not share any first-order variables apart from possibly x.

Purify$^+$
$$\frac{N}{N\left\{^{P(u)}/_{\top}\right\}}$$

side conditions:

(i) N is positive with respect to the minimal second-order variable.

Purify$^-$
$$\frac{N}{N\left\{^{P(u)}/_{\bot}\right\}}$$

side conditions:

(i) N is negative with respect to the minimal second-order variable.

$\not\approx$-Eliminate
$$\frac{N \cup \{x \not\approx a \vee \alpha(x)\}}{N \cup \{\alpha\,\{x/a\}\}}$$

side conditions:

(i) α does not contain the minimal second-order variable $\mathbf{P}$.

Table 4.5: The SOQE-calculus (3/4)

$\wedge$-Introduce
$$\frac{N \cup \{\alpha_1, \alpha_2\}}{N \cup \{\alpha_1 \wedge \alpha_2\}}$$

side conditions:

(i) α_1 and α_2 do not contain the minimal second-order variable $\mathbf{P}$,

(ii) $\not\approx$**-Eliminate** is not applicable.

$\forall$-Introduce
$$\frac{N \cup \{\alpha(x)\}}{N \cup \{\forall x \ \alpha\}}$$

where x is a free variable of α.
side conditions:

(i) α does not contain the minimal second-order variable $\mathbf{P}$,

(ii) $\not\approx$**-Eliminate** is not applicable.

$\exists$-Introduce
$$\frac{N \cup \{\alpha(c)\}}{N \cup \{\exists x \ \alpha \{{}^{c}/_{x}\}\}}$$

where a is a constant and x is a fresh variable.
side conditions:

(i) α does not contain the minimal second-order variable $\mathbf{P}$,

(ii) α does not contain any free variables,

(iii) c does not occus in N.

Table 4.6: The SOQE-calculus (4/4)

As with the MA-calculus, if the algorithm terminates successfully, we will have reduced $\left\{\exists x \ \neg NNF\Big(\pi(\Phi, x)\Big)\right\}$ to a set of clauses N free from second-order variables. Consequently, $\eta := \bigwedge N$ is a first-order formula. Again, we have that Φ is not valid if and only if $\Big\{\exists x \ \neg NNF\Big(\pi(\Phi, x)\Big)\Big\}$ is satisfiable, which is the case if and only if the first-order formula η is satisfiable, which is the case if and only is $\neg\eta$ is not valid. Hence, the frame validity of Φ corresponds to the first-order property $\neg\eta$.

If, however, the algorithm terminates having failed to eliminate all the second-order variables, we must choose a different ordering and repeat the process again. As with the MA-calculus, trying all orderings is no guarantee of success, but can, as Example 4.2.1 illustrates, make a difference in some situations.

Example 4.2.1. The SOQE-calculus fails to compute a first-order correspondence property for the frame validity of

$$\Box\left(\Box\neg p \leftrightarrow q\right) \rightarrow \Diamond\Box p \tag{4.2}$$

under the ordering $\mathbf{P} < \mathbf{Q}$, however, succeeds under the ordering $\mathbf{Q} < \mathbf{P}$. (Example taken from [23], where the polarity of p appears reversed.)

We start by taking the standard translation of the negation of (4.2), with the second-order variables all being implicitly existentially quantified:

$$\exists x \left[\forall y\Big(Rxy \rightarrow [\forall z\,(Ryz \rightarrow \neg\mathbf{P}z) \leftrightarrow \mathbf{Q}y]\Big) \wedge \neg\exists w\Big(Rxw \wedge \forall v\,[Rwv \rightarrow \mathbf{P}v]\Big)\right] .$$

We then convert this to the negation normal form

$$\exists x \left[\forall y\Big(\neg Rxy \vee \Big(\,[\exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y] \wedge [\forall z\,(\neg Ryz \vee \neg\mathbf{P}z) \vee \neg\mathbf{Q}y]\Big)\Big) \right.$$
$$\left. \wedge\forall w\Big(Rxw \rightarrow \exists v\,[Rwv \wedge \neg\mathbf{P}v]\Big)\right] ,$$

and proceed to non-deterministically apply the rules of Tables 4.3–4.6 as follows:

~~1.~~ $\exists x \Big[\forall y\Big(\neg Rxy \vee \big([\exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y] \wedge [\forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y] \big)\Big) \wedge \forall w\Big(Rxw \to \exists v\,[Rwv \wedge \neg \mathbf{P}v]\Big)\Big]$

~~2.~~ $\forall y\Big(\neg Ray \vee \big([\exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y] \wedge [\forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y] \big)\Big) \wedge \forall w\Big(Raw \to \exists v\,[Rwv \wedge \neg \mathbf{P}v]\Big)$ ∃-Eliminate, 1

~~3.~~ $\forall y\,\Big(\neg Ray \vee \big([\exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y] \wedge [\forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y] \big)\Big)$ ∧-Eliminate, 2

4. $\forall w\Big(Raw \to \exists v\,[Rwv \wedge \neg \mathbf{P}v]\Big)$ ∧-Eliminate, 2

~~5.~~ $\neg Ray \vee \big([\exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y] \wedge [\forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y] \big)$ ∀-Eliminate, 3

~~6.~~ $\neg Ray \vee \exists z\,(Ryz \wedge \mathbf{P}z) \vee \mathbf{Q}y$ ∧-Eliminate, 5

7. $\neg Ray \vee \forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y$ ∧-Eliminate, 5

8. $\neg Ray \vee \exists w\,(Ryw \wedge \mathbf{P}w) \vee \mathbf{Q}y$ Variable renaming, 6

The algorithm now terminates without eliminating $\mathbf{P}$ since we cannot apply the ∃-**Eliminate** rule due to the occurrence of the free variable y in clause 8.

Suppose, however, that we had used the ordering $\mathbf{Q} < \mathbf{P}$. Our algorithm would proceed as above to produce the set of clauses $\{4, 7, 8\}$ but, in this case, the algorithm is able to continue with its execution:

4. $\forall w\Big(Raw \to \exists v\,[Rwv \wedge \neg \mathbf{P}v]\Big)$

~~7.~~ $\neg Ray \vee \forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg \mathbf{Q}y$

~~8.~~ $\neg Ray \vee \exists w\,(Ryw \wedge \mathbf{P}w) \vee \mathbf{Q}y$

9. $\neg Ray \vee \forall z\,(\neg Ryz \vee \neg \mathbf{P}z) \vee \neg Ray \vee \exists w\,(Ryw \wedge \mathbf{P}w)$ Ackermann 8 into 7

We see that clause 9 contains both $\forall z(\neg Ryz \vee \neg \mathbf{P}z)$ and it's negation (up to variable renaming) $\exists w(Ryw \wedge \mathbf{P}w)$, and is, therefore, semantically equivalent to $\top$. Hence we may apply the **Simplification** rule, which then allows us to eliminate the remaining occurrences of $\mathbf{P}$.

10.	$\top$	Simplification, 9
~~11.~~	$\forall w\Big(Raw \to \exists v\,[Rwv \land \top]\Big)$	Purify^{-}, 4
~~12.~~	$\forall w\,(Raw \to \exists v Rwv)$	Simplification, 11
13.	$\exists x \forall w\,(Rxw \to \exists v Rwv)$	$\exists$-Introduce, 12

We conclude that the negation of (4.2) is satisfiable if and only if the first-order formula $\exists x \forall y\,(Rxy \to \exists z Ryz)$ is satisfiable. Consequently, we deduce that the frame validity of (4.2) corresponds to the first-order formula $\forall x \exists y\,(Rxy \land \forall z \neg Ryz)$.

4.3 The Sahlqvist-van Benthem Method

The Sahlqvist-van Benthem method is an algorithm introduced by Sahlqvist [21] as a means of demonstrating the first-order definability of correspondence properties for Sahlqvist formulas. The method was discussed as a practical means of computing correspondence properties by van Benthem in [24].

We describe here the version given in [5] (pages 156–162), and will make heavy use of the lambda notation $\lambda u.\ \varphi(u)$ which defines the unary predicate whose truth valuation coincides with the truth valuation of φ. That is to say, for a given model $\mathscr{N} = (W, *, V)$,

$$V\Big(\lambda u.\ \varphi(u)\Big) = \Big\{w \in W \quad \big| \quad \mathscr{N}\left(\begin{smallmatrix} x \\ w \end{smallmatrix}\right) \models \varphi(x)\Big\}\ ,$$

where $\mathscr{N}\left(\begin{smallmatrix} x \\ w \end{smallmatrix}\right)$ is the x-equivalent model with valuation $V\left(\begin{smallmatrix} x \\ w \end{smallmatrix}\right)$.

The algorithm comprises of five steps: the first three steps manipulate the input formula Φ until we have a form from which we can read off the minimal instances of second-order variables, which is step four. Step five is the instantiation step, where we eliminate all occurrences of the second-order variables, leaving us with a first-order formula corresponding to the frame validity of Φ.

Step 1: Reduce to a Sahlqvist Implication and Translate

Given a Sahlqvist formula Φ of $\mathscr{L}(\Diamond)$ we can, as a consequence of Proposition 3.3.1, effectively compute a Sahlqvist implication that is semantically equivalent to Φ. Hence, we may restrict our attention, without loss of generality, to the class of Sahlqvist implications.

We use the rule of Table 4.7 to distribute the main implication over any disjunctions in the antecedent of the Sahlqvist implication. We consider each conjunct Φ' in turn and take the conjunction of the individual correspondence properties.

input formula	*equivalent formula*
$(\alpha_1 \vee \alpha_2) \to \beta$	$(\alpha_1 \to \beta) \wedge (\alpha_2 \to \beta)$

Table 4.7: Rules for distributing implication over disjunction.

We now take the standard translation Ψ of the frame validity of each Φ'.

Step 2: Pull out Existential Quantifiers

Since Φ' is a Sahlqvist implication, the standard translation Ψ is of the form

$$\forall \mathbf{P}_1, \ldots, \forall \mathbf{P}_n \forall x \Big[Ant \to POS \Big] \ ,$$

where Ant is the standard translation of some Sahlqvist antecedent and POS is some positive formula. The formula Ant will, in general, contain existential quantifiers but, due to the syntactic definition of Sahlqvist antecedents, no existential quantifier occurs in the scope of a (first-order) universal quantifier.

Hence, using the rules in Table 4.8, we may pull out the existential quantifiers from the antecedent to the front of Ψ. We note that, in doing so, they become universal quantifiers due to the implicit negation of the main implication. We also pull the $\forall x$ quantifier to the front block of first-order quantifiers. It will be convenient to rename the variable x as x_0.

input formula	*equivalent formula*
$(\exists x_i \ \alpha(x_i)) \wedge \beta$	$\exists x_i \, [\alpha(x_i) \wedge \beta]$, where x_i does not occur free in β
$(\exists x_i \ \varphi) \to \psi$	$\forall x_i \, [\varphi \to \psi]$
$\forall \mathbf{P} \forall x_i \ \psi$	$\forall x_i \forall \mathbf{P} \ \psi$

Table 4.8: Rules for pulling out existential quantifiers.

We, thus, reduce Ψ to the form

$$\forall x_0 \forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge BOX \wedge NEG) \rightarrow POS \Big] ,$$

where REL is a conjunction of atomic formulas of the form Rx_ix_j, BOX is a conjunction of translations of boxed atomic formulas of the form $\forall y_k\,(Rx_iy_k \rightarrow \mathbf{P}_jy_k)$ and NEG is a conjunction of negative formulas.

We remark that by definition, translations of atomic formulas of the form $\mathbf{P}_jx_i$ are also considered to be boxed atomic formulas, with an empty string of boxes. For consistency we replace all formulas of the form $\mathbf{P}_jx_i$ with the equivalent form $\forall y_k\,(x_i \approx y_k \rightarrow \mathbf{P}_jy_k)$.

Step 3: Move Negative Formulas across the Implication

We move all negative formulas across the main implication using the rule in Table 4.9 to get Ψ into the form:

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge BOX) \rightarrow (\neg NEG \vee POS) \Big] ,$$

where $(\neg NEG \vee POS)$ is positive. Since we only require the main consequent to be positive, let us, henceforth, use the abbreviation $POS^* := \neg NEG \vee POS$.

input formula	*equivalent formula*
$(\alpha \wedge \beta) \rightarrow \psi$	$\alpha \rightarrow (\neg\beta \vee \psi)$

Table 4.9: Rule for moving negative formulas

Step 4: Read off Instances

We are now in a position to read off the minimal instances of $\mathbf{P}_i$ for each $1 \leq i \leq n$.

We have that every conjunct of BOX is either of the form $\forall y(x_i \approx y \to \mathbf{P}_j y)$ or of the form

$$\forall y_1 \Big(Rx_i y_1 \to \forall y_2 \Big(Ry_1 y_2 \to \ldots \forall y_k (Ry_{k-1} y_k \to \mathbf{P}_j y_k) \ldots \Big) \Big) , \qquad (4.3)$$

for some $\mathbf{P}_j$.

If η is a conjunct of BOX of the form $\forall y(x_i \approx y \to \mathbf{P}_j y)$, then η is already in PIA-form, whereas if η is of the form (4.3) then we can use the rules of Table 4.10 to rewrite η as the PIA-condition

$$\forall y_k \Big(\exists y_1 \exists y_2 \ldots \exists y_{k-1} (Rx_i y_1 \wedge Ry_1 y_2 \wedge \ldots \wedge Ry_{k-1} y_k) \to \mathbf{P}_j y_k \Big) ,$$

with $\mathbf{P}_j y_k$ occurring as the head and $\exists \vec{y}(Rx_i y_1 \wedge \ldots \wedge Ry_{k-1} y_k)$ as the tail. It will sometimes be convenient to use the abbreviation $R^k x_i y_k$ in lieu of the subformula $\exists \vec{y}(Rx_i y_1 \wedge \ldots \wedge Ry_{k-1} y_k)$.

input formula	*equivalent formula*
$\forall y_1 \, [\alpha(y_1) \to \forall y_2 \beta(y_2)]$	$\forall y_1 \forall y_2 \, [\alpha(y_1) \to \beta(y_2)]$
$\alpha \to (\beta \to \gamma)$	$(\alpha \wedge \beta) \to \gamma$
$\forall y_1 \forall y_2 \;\, \psi$	$\forall y_2 \forall y_1 \;\, \psi$
$\forall y_1 \forall y_2 \, [\varphi \to \psi]$	$\forall y_2 \, [(\exists y_2 \;\, \varphi) \to \psi]$
$\forall y \, (\varphi_1 \to \psi) \wedge \forall y \, (\varphi_2 \to \psi)$	$\forall y \, [(\varphi_1 \vee \varphi_2) \to \psi]$

Table 4.10: Rules for writing translated universal boxed forms into PIA-form

We construct the set

$$\Sigma_i = \left\{ \tau(u) \;\middle|\; \text{if } \forall u[\tau(u) \to \mathbf{P}_i u] \text{ is the PIA-form for some conjunct of } BOX \right\} ,$$

so that the minimal instances of $\mathbf{P}_i$ are given by the substitution,

$$\sigma(\mathbf{P}_i) = \lambda u. \; \bigvee \Sigma_i \; .$$

We note that, when Σ_i is the empty set, which corresponds to the case where Φ is positive with respect to $\mathbf{P}_i$, $\sigma(\mathbf{P}_i)$ is simply $\lambda u.\ u \not\approx u$ by convention.

Step 5: Instantiate

The final step of the algorithm is to apply the instantiation of $\mathbf{P}_i$ for each $1 \leq i \leq n$.

Since all of the $\sigma(\mathbf{P}_i)$ are free of second-order variables, we can uniformly substitute $\sigma(\mathbf{P}_i)$ for $\mathbf{P}_i$ for each $1 \leq i \leq n$. The substitutions have no effect on the second-order variable-free REL, and the result of applying the substitutions to BOX is tautologous since whenever $\mathbf{P}_i$ occurs in some boxed formula, by construction, one of the disjuncts in Σ_i is precisely the antecedent of this boxed formula.

The resulting formula is then

$$\forall x_1 \ldots \forall x_m \left[REL \rightarrow POS^* \left\{{}^{(\mathbf{P}_1,\ldots,\mathbf{P}_n)}/_{(\sigma(\mathbf{P}_1),\ldots,\sigma(\mathbf{P}_n))}\right\}\right] ,$$

where the substitution $\left\{{}^{(\mathbf{P}_1,\ldots,\mathbf{P}_n)}/_{(\sigma(\mathbf{P}_1),\ldots,\sigma(\mathbf{P}_n))}\right\}$ is the result of uniformly replacing each occurrence of $\mathbf{P}_i$ with $\sigma(\mathbf{P}_i)$ for each $1 \leq i \leq n$. This formula is now the desired first-order correspondent of Φ'.

To illustrate this algorithm we give the following examples:

Example 4.3.1. The axiom of Geach $\Diamond\Box p \rightarrow \Box\Diamond p$ corresponds to the first-order property of convergence [13] (page 126); that is, to

$$\forall x \forall y \forall w \Big[(Rxy \wedge Rxw) \rightarrow \exists v(Rwv \wedge Ryv)\Big] .$$

- In Step 1, we take the standard translation of of the frame validity of $\Diamond\Box p \rightarrow \Box\Diamond p$, that is, with all free variables universally quantified:

$$\forall \mathbf{P} \forall x \left[\exists y\Big(Rxy \wedge \forall z[Ryz \rightarrow \mathbf{P}z]\Big) \rightarrow \forall w\Big(Rxw \rightarrow \exists v[Rwv \wedge \mathbf{P}v]\Big)\right] .$$

- In Step 2 we pull out the existential quantifiers from the antecedent to get

$$\forall x\forall y\forall \mathbf{P}\Big[\Big(\underbrace{Rxy}_{REL}\wedge\underbrace{\forall z[Ryz\to\mathbf{P}z]\Big)}_{BOX}\to\underbrace{\forall w\Big(Rxw\to\exists v[Rwv\wedge\mathbf{P}v]\Big)}_{POS}\Big] .$$

- Step 3 is redundant since the conjunction NEG is empty.
- In Step 4, we read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.\ Ryu$.
- In Step 5, we apply the instantiation by substituting $\mathbf{P}$ with $\sigma(\mathbf{P})$ to give us

$$\forall x\forall y\Big[\Big(Rxy\wedge\forall z[Ryz\to Ryz]\Big)\to\forall w\Big(Rxw\to\exists v[Rwv\wedge Ryv]\Big)\Big] ,$$

which reduces to convergence.

Example 4.3.2. The axiom $\mathbf{alt}_1 = \Diamond p\to\Box p$ corresponds to functionality [13] (page 126); that is, to $\forall x\forall y\forall z\Big[(Rxy\wedge Rxz)\to y\approx z\Big]$.

- In Step 1, we take the standard translation of of the frame validity of $\Diamond p\to\Box p$:

$$\forall\mathbf{P}\forall x\Big[\exists y(Rxy\wedge\mathbf{P}y)\to\forall z(Rxz\to\mathbf{P}z)\Big] .$$

- In Step 2, we pull out all existential quantifiers from the antecedent to give us

$$\forall x\forall y\forall\mathbf{P}\Big[(\underbrace{Rxy}_{REL}\wedge\underbrace{\mathbf{P}y}_{BOX})\to\underbrace{\forall z(Rxz\to\mathbf{P}z)}_{POS}\Big] .$$

- Again, Step 3 is redundant since the conjunction NEG is empty.
- In Step 4, we read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.\ u\approx y$.
- In Step 5, we instantiate, replacing $\mathbf{P}$ with $\sigma(\mathbf{P})$ to give us

$$\forall x\forall y\Big[(Rxy\wedge y\approx y)\to\forall z(Rxz\to z\approx y)\Big] ,$$

which reduces to functionality.

We now show how the instantiation process of the Sahlqvist-van Benthem algorithm is exactly the substitution applied in those algorithms based on Ackermann's Lemma. In essence, the only difference between the two is that Ackermann's Lemma considers the second-order variables to be existentially quantified, whereas the instantiation approach acts of second-order variables that are universally quantified.

Proposition 4.3.1. *The correctness of the instantiation used in Sahlqvist-van Benthem algorithm can be deduced from the correctness of Ackermann's Lemma.*

Proof: Suppose we have completed Steps 1-4 of the algorithm and have our Sahlqvist implication Φ in the form:

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge BOX) \rightarrow POS^* \Big] ,$$

where BOX is a conjunction of translations of boxed atoms, REL a conjunction of relational formulas and POS^* is positive, together with a sets $\Sigma_1, \ldots, \Sigma_n$, where

$$\Sigma_i = \left\{ \tau(u) \;\middle|\; \text{ if } \forall u[\tau(u) \rightarrow \mathbf{P}_i u] \text{ is the PIA-form for some conjunct of } BOX \right\} .$$

Using the rules of Table 4.10, we can replace each conjunct of BOX as a semantically equivalent PIA-condition, so that we have Φ in the form

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge PIA) \rightarrow POS^* \Big] ,$$

where PIA is a conjunction of PIA-conditions. Furthermore, we can assume, without loss of generality, that there is precisely one PIA-condition with $\mathbf{P}_i$ occurring as the head. If there is no such PIA-condition for $\mathbf{P}_i$, then we may insert the tautologous PIA-condition $\forall u[u \not\approx u \rightarrow \mathbf{P}_i u]$ and if there are two or more such PIA-conditions for $\mathbf{P}_i$, then, by the rules of Table 4.10, we can join any two PIA-conditions $\forall u[\tau_1(u) \rightarrow \mathbf{P}_i u]$ and $\forall u[\tau_2(u) \rightarrow \mathbf{P}_i u]$ into the single PIA-condition $\forall u[(\tau_1 \vee \tau_2)(u) \rightarrow \mathbf{P}_1 u]$.

For convenience, let us write PIA_i for the unique PIA-condition with $\mathbf{P}_i$ occurring as the head and $\widehat{PIA}_i$ for the conjunction $\bigwedge_{j \neq i} PIA_j$.

We now select any $\mathbf{P}_i$ to eliminate and focus on the conjunct PIA_i in the subformula given below.

$$\forall \mathbf{P}_i \left[\forall u \left(\bigvee_{j=1}^{k} \tau_{i,j}(u) \rightarrow \mathbf{P}_i u \right) \wedge REL \wedge \widehat{PIA}_i \rightarrow POS^* \right]$$

We negate this formula to give us,

$$\exists \mathbf{P}_i \left[\forall u \left(\bigvee_{j=1}^{k} \tau_{i,j}(u) \to \mathbf{P}_i u \right) \wedge REL \wedge \widehat{PIA}_i \wedge \neg POS^* \right] ,$$

where we note that $\mathbf{P}_i$ does not occur in $\widehat{PIA}_i$ and, hence, we have that $REL \wedge \widehat{PIA}_i \wedge \neg POS^*$ is negative with respect to $\mathbf{P}_i$. In this form we may apply Ackermann's Lemma, the substitution being $\left\{ {}^{\mathbf{P}_i u}/_{\bigvee_{j=1}^{k} \tau_{i,j}(u)} \right\}$.

We note now that this is precisely the same as the instantiation given by the Sahlqvist-van Benthem method since the disjuncts $\tau_{i,j}$ are, by definition, precisely the elements of Σ_i. ∎

Chapter 5

Relationships between the Algorithms

We discuss, here, the similarities and differences in the computational capabilities of the three algorithms with regards to certain syntactic classes of basic modal axioms, which we introduced in Section 5.1, as well as their applcability to languages other than the basic modal language.

5.1 Application to the Sahlqvist Classes

We introduced the SOQE-calculus as a second-order version of the MA-calculus, but what do we actually mean by that? Put simply, each rule of the MA-calculus corresponds to a rule or a sequence of rules of SOQE. However, in order to highlight the relationship between these two algorithms, we make a small adaptation to the $\exists$-**Introduce** rule of SOQE, replacing it with the following:

∃-Introduce*
$$\frac{N \cup \{\alpha(a)\}}{N \cup \{\exists x \ (\alpha\,\{^{a}/_{x}\} \wedge x \approx a\}}$$
where a is a constant and x is a fresh variable.
side conditions:

(i) α does not contain the minimal second-order variable **P**

(ii) α does not contain any free variables.

The purpose of this adaptation is to allow us weaker side conditions for introducing existential quantifiers. We note that before, the rule was only applicable when a was a constant not occurring in N. Here we do not require such a restriction.

Theorem 5.1.1. *If N/M is a rule of the MA-calculus then there is a finite sequence of rules of the SOQE-calculus which reduces $\pi(N,x)$ to $\pi(M,x)$.*

Proof: Suppose that N/M is the **Surfacing** rule, then we take the standard translation of the main premise, $\alpha \vee \Box\beta$, of N/M, and apply the following sequence of rules:

~~1.~~	$\pi(\alpha,x) \vee \forall y\Big(\neg Rxy \vee \pi(\beta,y)\Big)$	
~~2.~~	$\pi(\alpha,x) \vee \neg Rxy \vee \pi(\beta,y)$	∀-Eliminate, 1
~~3.~~	$\pi(\alpha,z) \vee \neg Rzy \vee \pi(\beta,y)$	Variable Renaming, 2
~~4.~~	$\pi(\alpha,z) \vee \neg Rzx \vee \pi(\beta,x)$	Variable Renaming, 3
~~5.~~	$\pi(\alpha,y) \vee \neg Ryx \vee \pi(\beta,x)$	Variable Renaming, 4
~~6.~~	$\forall y\Big(\pi(\alpha,y) \vee \neg Ryx\Big) \vee \pi(\beta,x)$	∀-Introduce, 5
7.	$\forall y\Big(\neg Ryx \vee \pi(\alpha,y)\Big) \vee \pi(\beta,x)$	Simplification, 6

This is now just the standard translation of the main consequent, $\Box^{\smile}\alpha \vee \beta$, of N/M, as required.

Suppose that N/M is the **Skolemization** rule, then we take the standard translation of $\neg a \vee \Diamond\beta$ and apply the following sequence of rules:

~~1.~~	$x \not\approx a \vee \exists y\Big(Rxy \wedge \pi(\beta, y)\Big)$	
~~2.~~	$\exists y\Big(Ray \wedge \pi(\beta, y)\Big)$	$\not\approx$-Eliminate, 1
~~3.~~	$Rab \wedge \pi(\beta, b)$	$\exists$-Eliminate, 2
~~4.~~	Rab	$\wedge$-Eliminate, 3
~~5.~~	$\pi(\beta, b)$	$\wedge$-Eliminate, 3
~~6.~~	$\exists y(Ray \wedge y \approx b)$	$\exists$-Introduce*, 4
7.	$x \not\approx a \vee \exists y(Rxy \wedge y \approx b)$	Term abstraction, 6
8.	$x \not\approx b \vee \pi(\beta, x)$	Term abstraction, 5

It is here, in clause 6, where we have made use of the adapted rule for introducing existential quantifiers. The remaining clauses are the standard translations of the main consequents $\neg a \vee \Diamond b$ and $\neg b \vee \beta$, as required.

Suppose that N/M is the **Clausify** rule, then the main premise is $\neg(\alpha \vee \beta)$. As before, we take the standard translation and apply the following sequence of rules:

~~1.~~	$\neg\Big(\pi(\alpha, x) \vee \pi(\beta, x)\Big)$	
~~2.~~	$\neg\pi(\alpha, x) \wedge \neg\pi(\beta, x)$	Simplification, 1
3.	$\neg\pi(\alpha, x)$	$\wedge$-Eliminate, 2
4.	$\neg\pi(\beta, x)$	$\wedge$-Eliminate, 2

Here, clauses 3 and 4 are exactly the standard translations of the main consequents.

The remaining cases are where N/M is the **Ackermann** rule or one of the **Purify** rules. These correspond directly to the **Ackermann** and **Purify** rules of the SOQE-calculus without the need for any sequence of manipulations.

Hence, we have shown that any rule of the MA-calculus can be emulated by a sequence of rules of the SOQE-calculus. ∎

We note that the adaptation of the **∃-Introduce** rule, although convenient for demonstrating the relationship between the two algorithms, is unnecessary since we only apply it to formulas containing no second-order variables. No rules of the MA-calculus are applicable to formulas of the form $\neg a \vee \Diamond b$ and nor are any rules of the SOQE-calculus applicable to formulas of the form Rab where b occurs elsewhere in N. Hence, we have the following corollary.

Corollary 5.1.1. *If the MA-calculus successfully reduces the modal formula φ to a formula, free of propositional variables, under the ordering $p_1 < p_2 < \cdots < p_n$ then the SOQE-calculus successfully reduces $\pi(\varphi, x)$ to a first-order formula, under the ordering $\boldsymbol{P}_1 < \boldsymbol{P}_2 < \cdots < \boldsymbol{P}_n$.*

Proof: This is an immediate consequence of the above Theorem, since any finite sequence of rules of the MA-calculus can be emulated by a finite sequence of rules of the SOQE-calculus and every second-order variable of $\pi(\varphi, x)$ corresponds to some propositional variable of φ. ∎

In particular, since the MA-calculus can be used to successfully compute correspondence properties for the class of generalized Sahlqvist formulas (see Theorem 4.1.1), so too can the SOQE-calculus.

By contrast, the Sahlqvist-van Benthem method is a specific procedure only applicable to Sahlqvist formulas; as so, is not as general as the MA-calculus and the SOQE-calculus since it cannot be applied to generalized Sahlqvist formulas.

We summarize this in Table 5.1.

	MA	*SOQE*	*S-vB method*
Sahlqvist class	✓	✓	✓
generalized Sahlqvist class	✓	✓	✗
simply generalized Sahlqvist class	✗	✗	✗

Table 5.1: Table indicating the success (✓) or failure (✗) of our algorithms for various classes.

5.2 Application to other Languages

Although we have not specified it for such purposes, the MA-calculus is well suited for computing first-order correspondence properties for some formulas of $\mathscr{L}_n^{\smile}(\Diamond)$. For example:

Example 5.2.1. The frame validity of the $\mathscr{L}_n^{\smile}(\Diamond)$-formula $\Box p \to \Diamond^{\smile} p$ corresponds to $\forall x \exists y(Rxy \wedge Ryx)$.

We apply the rules of the MA-calculus to the set $\{\neg a \vee (\Box p \wedge \Box^{\smile}\neg p)\}$ as follows:

~~1.~~	$\neg a \vee (\Box p \wedge \Box^{\smile}\neg p)$	
~~2.~~	$\neg a \vee \Box p$	Clausify, 1
~~3.~~	$\neg a \vee \Box^{\smile}\neg p$	Clausify, 2
~~4.~~	$\Box^{\smile}\neg a \vee p$	Surfacing, 2
5.	$\neg a \vee \Box^{\smile}\Box^{\smile} a$	Ackermann, 4 into 3

Here, the algorithm terminates having successfully eliminated all propositional variables. We now take the standard translation, which simplifies to

$$\forall y\Big(Rya \to \forall z(Rzy \to z \not\approx a)\Big) .$$

We unskolemize this to give us

$$\exists x \forall y\Big(Ryx \to \forall z(Rzy \to z \not\approx x)\Big) ,$$

which corresponds to a refutation of the validity of $\Box p \to \Diamond^{\smile} p$. Therefore, negating this, we get that $\forall x \exists y(Ryx \wedge Rxy)$ is the first-order correspondent of $\Box p \to \Diamond^{\smile} p$.

Since the SOQE-calculus and the Sahlqvist-van Benthem method both operate directly on the standard translations and, since the adjustment from Rxy to Ryx in the standard translations of $\Diamond\varphi$ and $\Diamond^{\smile}\varphi$ makes little difference to the procedures, these two algorithms can also be used to compute correspondence properties of certain $\mathscr{L}_n^{\smile}(\Diamond)$-formulas.

In fact, this observation that certain changes like this go unnoticed by the SOQE-calculus and the Sahlqvist-van Benthem method means we can apply these algorithms to formulas of $\mathscr{L}(\Diamond, D)$ without issue, as the following example illustrates:

Example 5.2.2. The frame validity of the $\mathscr{L}(\Diamond, D)$-formula $\left(\neg p \wedge \overline{D}p\right) \rightarrow \Box p$ corresponds to irreflexivity.

We start by taking the standard translation of the validity of $\left(\neg p \wedge \overline{D}p\right) \rightarrow \Box p$:

$$\forall \mathbf{P} \forall x \quad \left[\underbrace{\neg \mathbf{P}x}_{NEG} \wedge \underbrace{\forall y \left(x \not\approx y \rightarrow \mathbf{P}y\right)}_{BOX} \rightarrow \underbrace{\forall z \left(Rxz \rightarrow \mathbf{P}z\right)}_{POS} \right] .$$

Moving NEG across to the consequent of the main implication, we can apply the substitution $\sigma(\mathbf{P}) = \lambda u.\ x \not\approx u$ since $x \not\approx y$ occurs as the tail of the boxed formula and y is the quantified variable.

Hence, we have that $\left(\neg p \wedge \overline{D}p\right) \rightarrow \Box p$ is equivalent to the first-order formula

$$\forall x \Big[\forall y \left(x \not\approx y \rightarrow x \not\approx y\right) \rightarrow \Big(x \not\approx x \vee \forall z \left(Rxz \rightarrow x \not\approx z\right) \Big) \Big] ,$$

which reduces to $\forall x \neg Rxx$, as required.

Alternatively, we can apply the rules of the SOQE-calculus to the set

$$\Big\{ \exists x \Big[\neg \mathbf{P}x \wedge \forall y \left(x \approx y \vee \mathbf{P}y\right) \wedge \exists z \left(Rxz \wedge \neg \mathbf{P}z\right) \Big] \Big\} ,$$

which corresponds to a refutation of the validity of $\left(\neg p \wedge \overline{D}p\right) \rightarrow \Box p$.

We have the following computation:

~~1.~~	$\neg \mathbf{P}a \wedge \forall y\,(a \approx y \vee \mathbf{P}y) \wedge \exists z\,(Raz \wedge \neg \mathbf{P}z)$	$\exists$-Eliminate
~~2.~~	$\neg \mathbf{P}a$	$\wedge$-Eliminate, 1
~~3.~~	$\forall y\,(a \approx y \vee \mathbf{P}y)$	$\wedge$-Eliminate, 1
~~4.~~	$\exists z\,(Raz \wedge \neg \mathbf{P}z)$	$\wedge$-Eliminate, 1
~~5.~~	$a \approx y \vee \mathbf{P}y$	$\forall$-Eliminate, 3
~~6.~~	$\neg\neg a \approx a$	Ackermann 5 into 2
~~7.~~	$\exists z\,(Raz \wedge \neg\neg a \approx z)$	Ackermann 5 into 4
8.	$\top$	Simplification, 6
~~9.~~	Raa	Simplification, 7
10.	$\exists x Rxx$	$\exists$-Introduce, 9

Negating this, we have that the frame validity of $\big(\neg p \wedge \overline{D}p\big) \rightarrow \Box p$ corresponds to irreflexivity.

Both algorithms can be applied to all the formulas of Table 3.6 to compute their respective correspondence properties. See Appendix A for full details.

Clearly, in this respect, the SOQE-calculus is a far more versatile algorithm. Furthermore, it is far more versatile than the Sahlqvist-van Benthem method, which, again, is restricted to the class of Sahlqvist formulas, whereas the SOQE-calculus can be used to calculate the correspondence property for $Dp \wedge \overline{D}(p \rightarrow q) \rightarrow \overline{D}q$, which stipulates that universe contains at most two elements.

Example 5.2.3. We apply the SOQE-calculus to the clause set

$$\Big\{\exists x\Big(\exists y[x \not\approx y \wedge \mathbf{P}y] \wedge \forall z[x \approx z \vee \neg \mathbf{P}z \vee \mathbf{Q}z] \wedge \exists w[x \approx w \wedge \neg \mathbf{Q}w]\Big)\Big\}\ ,$$

which corresponds to a refutation of the validity of $Dp \wedge \overline{D}(p \rightarrow q) \rightarrow \overline{D}q$. We have the computation:

~~1.~~	$\exists y[a \not\approx y \wedge \mathbf{P}y] \wedge \forall z[a \approx z \vee \neg \mathbf{P}z \vee \mathbf{Q}z]$ $\wedge \exists w[a \approx w \wedge \neg \mathbf{Q}w]$	$\exists$-Eliminate
~~2.~~	$\exists y[a \not\approx y \wedge \mathbf{P}y]$	$\wedge$-Eliminate, 1
~~3.~~	$\forall z[a \approx z \vee \neg \mathbf{P}z \vee \mathbf{Q}z]$	$\wedge$-Eliminate, 1
~~4.~~	$\exists w[a \not\approx w \wedge \neg \mathbf{Q}w]$	$\wedge$-Eliminate, 1
~~5.~~	$a \approx z \vee \neg \mathbf{P}z \vee \mathbf{Q}z$	$\forall$-Eliminate, 3
~~6.~~	$\exists w[a \not\approx w \wedge (a \approx w \vee \neg \mathbf{P}w)]$	Ackermann, 3 into 4
~~7.~~	$\exists w[a \not\approx w \wedge \neg \mathbf{P}w]$	Simplification, 6
~~8.~~	$a \not\approx b \wedge \mathbf{P}b$	$\exists$-Eliminate, 2
9.	$a \not\approx b$	$\wedge$-Eliminate, 8
~~10.~~	$\mathbf{P}b$	$\wedge$-Eliminate, 8
~~11.~~	$x \not\approx b \vee \mathbf{P}x$	Term abstraction, 10
12.	$\exists w[a \not\approx w \wedge w \not\approx b]$	Ackermann, 11 into 7

It follows by a few more straightforward applications of the rules that a refutation of $Dp \wedge \overline{D}(p \to q) \to \overline{D}q$ corresponds to the first-order property $\exists x \exists y \exists z(x \not\approx y \wedge x \not\approx z \wedge y \not\approx z)$. Consequently, the frame validity of $Dp \wedge \overline{D}(p \to q) \to \overline{D}q$ corresponds to $\forall x \forall y \forall z(x \not\approx y \wedge y \not\approx z \to z \approx x)$, which stipulates that the universe contains at most two elements.

Table 5.2, below, indicates the applicability of each of the three algorithms to the various languages we discussed in Chapter 3.

	MA	*SOQE*	*S-vB method*
$\mathscr{L}(\Diamond)$	✓	✓	✓
$\mathscr{L}_n^{\smile}(\Diamond)$	✓	✓	✓
$\mathscr{L}(\Diamond, D)$	✗	✓	✓
$\mathscr{L}\big((\geq n) \mid n \in \mathbb{N}\big)$	✗	✗	✗

Table 5.2: Table indicating the applicability (✓) or non-applicability (✗) of our algorithms to various languages.

Chapter 6

Extensions

Since Ackermann's Lemma appears to be the key to the successful quantifier elimination necessary for computing the correspondence properties for the class of Sahlqvist and generalized Sahlqvist formulas, is it possible to generalize Ackermann's result and develop the tools for even greater automated correspondence procedures?

The following generalization was introduced by Nonnengart and Szałas in [20]:

Lemma 6.0.1. *As with Ackermann's lemma, let $\boldsymbol{P}$ be an n-ary second-order variable, $\vec{y}$ be an n-tuple of first-order variables and $\vec{x}$ another tuple of first-order variables. However, we now allow α to contain positive occurrences of the variable $\boldsymbol{P}$. Again, let NEG (resp. POS) be a formula which is negative (reps. positive) with respect to $\boldsymbol{P}$. We have the following equivalences:*

$$\exists \boldsymbol{P}\Big(\forall \vec{y}\,[\alpha(\boldsymbol{P}, \vec{y}, \vec{x}) \to \boldsymbol{P}(\vec{y})] \wedge NEG(\boldsymbol{P})\Big) \equiv NEG\left\{{}^{\boldsymbol{P}(\vec{u})}/_{[\text{LFP }\ \boldsymbol{P}.\ \alpha(\boldsymbol{P},\vec{y},\vec{x})](\vec{u})}\right\}$$

and

$$\exists \boldsymbol{P}\Big(\forall \vec{y}\,[\boldsymbol{P}(\vec{y}) \to \alpha(\boldsymbol{P}, \vec{y}, \vec{x})] \wedge POS(\boldsymbol{P})\Big) \equiv POS\left\{{}^{\boldsymbol{P}(\vec{u})}/_{[\text{GFP }\ \boldsymbol{P}.\ \alpha(\boldsymbol{P},\vec{y},\vec{x})](\vec{u})}\right\},$$

where $[\text{LFP } \boldsymbol{P}.\ \alpha(\boldsymbol{P}, \vec{y}, \vec{x})]\,(\vec{u})$ (resp. $[\text{GFP } \boldsymbol{P}.\ \alpha(\boldsymbol{P}, \vec{y}, \vec{x})]\,(\vec{u})$) is the least (resp. greatest) fixed-point of $\alpha(\boldsymbol{P}, \vec{y}, \vec{x})$ with respect to $\boldsymbol{P}$. See Chapter 2.3.

This result subsumes Lemma 4.0.1, for if α does not contain any occurrences of the second-order variable $\mathbf{P}$, then $[\text{LFP } \mathbf{P}.\ \alpha]\,(\vec{x}) \equiv \alpha(\vec{x})$ and $[\text{GFP } \mathbf{P}.\ \alpha]\,(\vec{x}) \equiv \alpha(\vec{x})$.

6.1 Extending the MA-Calculus

The language of modal fixed-points $\mathscr{L}^{\mu}(\Diamond)$ is considered in [25] and [4], where we allow formulas of the form $(\mu p\ \varphi)$, where φ is a modal formula, positive with respect to the propositional variable p.

We extend the standard translation with the rule

$$\pi(\mu p\ \varphi, x) = [\text{LFP}\ \mathbf{P}.\ \pi(\varphi, y)]\,(x)\ \text{, where } y \text{ is a fresh variable.}$$

With this translation we can reformulate Lemma 6.0.1 in the language of modal fixed-points.

Lemma 6.1.1. *Let p be a propositional variable and let α be a modal formula, positive with respect to p. Again, let NEG be a modal formula which is negative with respect to p. We have that $\mathscr{M} \models NEG\left\{{}^{p}/_{\mu p\ \alpha}\right\}$ if and only if $\mathscr{M}' \models (\alpha \rightarrow p) \wedge NEG(p)$ for some p-equivalent model $\mathscr{M}'$.*

Proof: Immediate from Lemma 6.0.1 and the definitions given in Section 3.2. ∎

With this lemma we can adapt the **Ackermann** rule of the MA-calculus, replacing it with the rule in table 6.1.

Ackermann*

$$\frac{N \cup \{\alpha_1(p) \vee p, \ldots, \alpha_n(p) \vee p\}}{N\left\{{}^{p}/_{\mu p\ (\neg\alpha_1 \vee \ldots \vee \neg\alpha_n)}\right\}}$$

side conditions:

(i) the minimal propositional variable p occurs only negatively in each α_i,

(ii) N is negative with respect to p.

Table 6.1: Additional rule for the MA-calculus

This extended algorithm is then able to deal with formulas such as Löb's axiom, which has no first-order correspondent ([5] pages 130–131).

Example 6.1.1. The frame validity of Löb's axiom $\Box(\Box p \to p) \to \Box p$ corresponds to the formula

$$\forall x \forall y \Big(Rxy \to [\text{LFP } \mathbf{P}.\ Rxu \wedge \forall z(Ruz \to \mathbf{P}z)]\,(y) \Big)$$

in the first-order logic of fixed-points [13] (pages 156–157).

We apply the extended MA-calculus to the clause set $\Big\{ \neg a \vee \Big(\Box(\Diamond p \vee p) \wedge \Diamond\neg p \Big) \Big\}$, which corresponds to a refutation of Löb's axiom, as follows:

~~1.~~	$\neg a \vee \Big(\Box(\Diamond\neg p \vee p) \wedge \Diamond\neg p \Big)$	
~~2.~~	$\neg a \vee \Box(\Diamond\neg p \vee p)$	Clausify, 1
~~3.~~	$\neg a \vee \Diamond\neg p$	Clausify, 1
~~4.~~	$\Box^{\smile}\neg a \vee \Diamond\neg p \vee p$	Surfacing, 2
5.	$\neg a \vee \Diamond\neg\mu p\ \neg(\Box^{\smile}\neg a \vee \Diamond\neg p)$	Ackermann*, 4 into 3

The algorithm now terminates, having reduced $\Big\{ \neg a \vee \Big(\Box(\Diamond p \vee p) \wedge \Diamond\neg p \Big) \Big\}$ to a set of clauses in which every propositional variable occurs bound by the μ-constructor. We now take the standard translation of clause 5 and simplify to get,

$$\exists y \Big(Ray \wedge \neg\, [\text{LFP } \mathbf{P}.\ Rau \wedge \forall z(Ruz \to \mathbf{P}z)]\,(y) \Big)\ .$$

We now unskolemize to give us

$$\exists x \exists y \Big(Rxy \wedge \neg\, [\text{LFP } \mathbf{P}.\ Rxu \wedge \forall z(Ruz \to \mathbf{P}z)]\,(y) \Big)\ ,$$

which corresponds to a refutation of the validity of Löb's axiom. Negating this, we have that the frame validity of Löb's axiom corresponds to

$$\forall x \forall y \Big(Rxy \to [\text{LFP } \mathbf{P}.\ Rxu \wedge \forall z(Ruz \to \mathbf{P}z)]\,(y) \Big)\ . \tag{6.1}$$

By a method, which we do not describe here (see [13], pages 34–35), we may characterize this correspondence property by recursively generating first-order approximations of (6.1). We find that, in doing this, Löb's axiom corresponds to the accessibility relation $\widehat{R}$ being transitive and reverse well-founded.

We note that Löb's axiom is not a generalized Sahlqvist formula, nor is it semantically equivalent to any generalized Sahlqvist formula since it does not have a first-order property. It is, however, a simply generalized Sahlqvist formula, as defined in Definition 3.3.1. The fact that this extended algorithm is well equipped to deal with Löb's axiom motivates the following Theorem.

Theorem 6.1.1. *The extended MA-calculus successfully computes correspondence properties of monadic regular formulas, in the first-order logic of fixed-points, for* any *ordering on the propositional variables.*

Proof: This proof is adapted from the proof for the monadic inductive class in [13] (page 180), where we have removed the condition that the dependency relation be acyclic. Suppose that α is a a monadic regular formula, then α is either a negated universal boxed form, a propositional constant, a positive formula of the form $\varphi \wedge \psi$, $\varphi \vee \psi$ or for some monadic regular formulas φ and ψ.

By repeated applications of the **Reduce** and **Clausify** rules of the MA-calculus, we can reduce the input clause $\neg a \vee \neg\alpha$ to a set containing clauses of the form

surfaced clauses:	$\beta \wedge p$
negative clauses:	β
universal boxed forms:	$\beta\wedge$
relational clauses:	$\neg b \vee \Diamond c$
existential clauses:	$\neg b \vee \Diamond\delta$

where b and c are nominals, p is a propositional variable, β is negative, γ is a universal boxed form, and δ is either a boxed form or a conjunction of positive formulas and boxed forms.

Since we are dealing with the class of monadic regular formulas, we may have propositional variables occurring both positively and negatively in surfaced clauses and it universal boxed forms. However, it follows for the definition that there can only be one positive occurrence of any variable in any surfaced clause of universal boxed form.

By considering each rule in turn, we see that this class of clauses is closed under all rules of the MA-calculus. For any ordering on the propositional variables, we can bring positive occurrences of the minimal propositional variable p to the surface using the **Surfacing** rule. And since there is a maximum of one positive occurrence of p in each clause we will be left with a form where we can apply Theorem 6.1.1 to eliminate p. This completes the proof. ∎

6.2 Extending the SOQE-Calculus

In a similar approach to the previous section, we can extend the SOQE-calculus by replacing the **Ackermann** rule with the rule given in Table 6.2.

Ackermann*

$$\frac{N \cup \{\alpha_1(\mathbf{P}, x) \vee \mathbf{P}x, \ldots, \alpha_n(\mathbf{P}, x) \vee \mathbf{P}x\}}{N\left\{{}^{\mathbf{P}u}\!/\![\mathrm{LFP}\ \mathbf{P}.\ (\sim\alpha_1 \vee \ldots \vee \sim\alpha_n)](u)\right\}}$$

side conditions:

(i) the minimal second-order variable $\mathbf{P}$ occurs only negatively in each α_i,

(ii) N is negative with respect to $\mathbf{P}$,

(iii) N and α_i do not share any of the same first-order variables. α_i and α_j do not share any first-order variables apart from possibly x.

Table 6.2: Extended rule for SOQE

Unsurprisingly, the extended SOQE-calculus produces the same correspondence property for the frame validity of Löb's axiom, as we now show.

Example 6.2.1. The standard translation of the negation of Löb's axiom (with second-order variables implicitly existentially quantified) is computed as

$$\exists x \neg \left[\forall y \Big(Rxy \to [\forall z\,(Ryz \to \mathbf{P}z) \to \mathbf{P}y]\Big) \to \forall w\,(Rxw \to \mathbf{P}w)\right] ,$$

which we then convert to the negation normal form:

$$\exists x \left[\forall y \Big(\neg Rxy \vee \exists z\,(Ryz \wedge \neg \mathbf{P}z) \vee \mathbf{P}y\Big) \wedge \exists w\,(Rxw \wedge \neg \mathbf{P}w)\right] .$$

We now proceed to apply the rules of the extended SOQE-calculus as follows:

~~1.~~	$\exists x \left[\forall y \Big(\neg Rxy \lor \exists z \left(Ryz \land \neg \mathbf{P}z\right) \lor \mathbf{P}y\Big) \land \exists w \left(Rxw \land \neg \mathbf{P}w\right)\right]$	
~~2.~~	$\forall y \Big(\neg Rxy \lor \exists z \left(Ryz \land \neg \mathbf{P}z\right) \lor \mathbf{P}y\Big) \land \exists w \left(Raw \land \neg \mathbf{P}w\right)$	$\exists$-Eliminate, 1
~~3.~~	$\forall y \Big(\neg Rxy \lor \exists z \left(Ryz \land \neg \mathbf{P}z\right) \lor \mathbf{P}y\Big)$	$\land$-Eliminate, 2
~~4.~~	$\exists w \left(Raw \land \neg \mathbf{P}w\right)$	$\land$-Eliminate, 2
~~5.~~	$\neg Rxy \lor \exists z \left(Ryz \land \neg \mathbf{P}z\right) \lor \mathbf{P}y$	$\forall$-Eliminate, 3
~~6.~~	$\exists w \Big(Raw \land \neg \left[\textsc{lfp}\ \mathbf{P}.\ Rxu \land \forall z \left(Ruz \to \mathbf{P}z\right)\right](y)\Big)$	Ackermann*, 5 into 4
7.	$\exists x \exists w \Big(Rxw \land \neg \left[\textsc{lfp}\ \mathbf{P}.\ Rxu \land \forall z \left(Ruz \to \mathbf{P}z\right)\right](y)\Big)$	$\exists$-Introduce, 6

The algorithm now terminates, having successfully eliminated all occurrences of $\mathbf{P}$ not bound by fixed-point constructors. Negating this, we have that the validity of Löb's axiom corresponds to

$$\forall x \forall w \Big(Rxw \to \left[\textsc{lfp}\ \mathbf{P}.\ Rxu \land \forall z \left(Ruz \to \mathbf{P}z\right)\right](y)\Big) ,$$

as expected.

As previously noted, Löb's axiom is not semantically equivalent to any generalized Sahlqvist formula but is a simply generalized Sahlqvist formula, which motivates the following Theorem.

Theorem 6.2.1. *The extended SOQE-calculus successfully computes the correspondence properties in the first-order logic of fixed-points for monadic regular formulas for* any *ordering on the propositional variables.*

Proof: We showed in Theorem 5.1.1 that every rule of the MA-calculus can be emulated be a similar sequence of rules of the SOQE-calculus acting on the standard translation. This additional rule is no exception and corresponds directly to the standard translation of the **Ackermann*** of the extended MA-calculus. We showed in Theorem 6.1.1 that the extended MA-calculus successfully computes the correspondence properties for all monadic regular formulas for any ordering on the propositional variables. Consequently, the extended SOQE-calculus must share this same property. ∎

6.3 Extending the Sahlqvist-van Benthem Method

We saw in Proposition 4.3.1 that the instantiations of the Sahlqvist-van Benthem method can be viewed as substitution instances of Ackermann's Lemma. Extending the domain of our algorithm to simply generalized Sahlqvist formulas, we can make full use of the generalization of Nonnengart and Szałas to compute correspondence properties for this larger class.

We proceed with steps 1-3 as before, to get a formula of the form

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge UNI) \to (\neg NEG \vee POS) \Big] ,$$

where, now, UNI is a conjunction of translations of universal boxed forms. We choose an ordering on the second-order variables $<$, and proceed as follows:

Step 4b: Read off Instances of the Minimal Variable

We note that every conjunct of UNI is either of the form $\forall y(x_i \approx y \to \mathbf{P}_j y)$ or of the form

$$\forall y_1 \Big(Rx_i y_1 \to \beta_1(y_1) \to \forall y_2 \Big(Ry_1 y_2 \to \ldots \\ \to \beta_{k-1}(y_{k-1}) \to \forall y_k (Ry_{k-1} y_k \to \beta_k(y_k) \to \mathbf{P}_j y_k) \ldots \Big)\Big) , \quad (6.2)$$

for some $\mathbf{P}_j$. If η is a conjunct of UNI of the form $\forall y(x_i \approx y \to \mathbf{P}_j y)$, then η is already in PIA-form, whereas if η is of the form (6.2) then we can use the rules of Table 4.10 to rewrite η as the PIA-condition

$$\forall y_k \Big(\exists y_1 \exists y_2 \ldots \exists y_{k-1} \Big(Rx_i y_1 \wedge Ry_1 y_2 \wedge \ldots \wedge Ry_{k-1} y_k \\ \wedge \beta_1(y_1) \wedge \ldots \wedge \beta_{k-1}(y_{k-1}) \Big) \wedge \beta_k(y_k) \to \mathbf{P}_j y_k \Big) .$$

Suppose that $\mathbf{P}_i$ is minimal with respect to $<$, then we construct the set

$$\Sigma_i = \left\{ \tau(u) \;\middle|\; \text{if } \forall u[\tau(u) \to \mathbf{P}_i u] \text{ is the PIA-form for some conjunct of } UNI \right\} ,$$

so that the minimal instances of $\mathbf{P}_i$ are given by the substitution,

$$\sigma(\mathbf{P}_i) = \lambda u. \quad \left[\text{LFP } \mathbf{P}_i. \ \bigvee \Sigma_i\right](u) \ .$$

This construction is possible since $\mathbf{P}_i$ occurs only positively in each β_j and, consequently, $\bigvee \Sigma_i$ is also positive with respect to $\mathbf{P}_i$.

As in the original algorithm, we note that when Σ_i is the empty set, $\sigma(\mathbf{P}_i)$ is simply $\lambda u.\ u \not\approx u$ by convention.

Step 5b: Instantiate

We now apply the instantiation for the minimal second-order variable $\mathbf{P}_i$.

We uniformly substitute $\sigma(\mathbf{P}_i)$ for $\mathbf{P}_i$. Again, the substitution has no effect on the second-order variable-free REL.

Applying the substitution to universal boxed forms, with $\mathbf{P}_i$ occurring as the head, results in a tautology. For suppose $\forall y[\tau \to \mathbf{P}_i]$ is a universal boxed form occurring in UNI, rewritten in PIA-form, then τ is a positive formula. The result of applying the instantiation is, therefore, equivalent to

$$\forall y\Big[\tau \left\{{}^{\mathbf{P}_i u}/_{[\text{LFP } \mathbf{P}_i.\ \bigvee \Sigma_i](u)}\right\} \to \left[\text{LFP } \mathbf{P}_i. \ \bigvee \Sigma_i\right](y)\Big] \ . \tag{6.3}$$

Suppose that $\mathscr{N}$ is some model of the antecedent of (6.3), then it follows that

$$\mathscr{N} \models \left(\bigvee \Sigma_i\right) \left\{{}^{\mathbf{P}_i u}/_{[\text{LFP } \mathbf{P}_i.\ \bigvee \Sigma_i](u)}\right\} \ ,$$

since this contains the antecedent as one of its disjuncts. Since $[\text{LFP } \mathbf{P}_i.\ \bigvee \Sigma_i](u)$ is a fixed-point, it follows from the definition, that

$$\mathscr{N} \models \left[\text{LFP } \mathbf{P}_i. \ \bigvee \Sigma_i\right](u) \ ,$$

which is to say that (6.3) is a tautology.

Applying the substitution to universal boxed forms, with $\mathbf{P}_i$ not occurring as the head, results in another universal boxed form, since positive occurrences of

$\mathbf{P}_i$ are replaced with $\sigma(\mathbf{P}_i)$ which is positive; we are not forced outside the class of universal boxed forms.

Step 6b: Repeat

Having successfully eliminated all occurrences of the minimal second-order variable $\mathbf{P}_i$, we still have the form

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge UNI) \rightarrow (\neg NEG \vee POS) \Big] \; ,$$

where the conjuncts of UNI and POS may, now, contain fixed-point constructs. However, by Proposition 2.3.1, we still have the result that positivity implies monotonicity, which we need for the instantiation to be successful.

Hence we can repeat Steps 4b and 5b until all second-order variables have been instantiated.

The resulting formula is then

$$\forall x_1 \ldots \forall x_m \left[REL \rightarrow POS^* \left\{ {}^{\mathbf{P}_1}/_{\sigma(\mathbf{P}_1)} \right\} \ldots \left\{ {}^{\mathbf{P}_n}/_{\sigma(\mathbf{P}_n)} \right\} \right] \; ,$$

which is the desired correspondent of Φ' in the first-order logic of fixed-points.

Example 6.3.1. We demonstrate the procedure for the formula given in Example 4.2.1 above, which, although does not have a correspondence property in terms of fixed-points, it does have a cyclic dependency relation $\succ_d$ on the universal boxed forms.

$$\Box (\Box \neg p \leftrightarrow q) \rightarrow \Diamond \Box p \; .$$

- In order to apply the algorithm we must first rewrite this as a semantically equivalent simply generalized Sahlqvist formula:

$$\Big[\Box (\Box \neg p \rightarrow q) \wedge \Box (q \rightarrow \Box \neg p) \Big] \rightarrow \Diamond \Box p \; ,$$

- In Step 1, we take the standard translation of the frame validity of this formula (that is, with all free-variables, universally quantified):

$$\forall \mathbf{P} \forall \mathbf{Q} \forall x \quad \Big[\forall y_0 \Big(Rxy_0 \wedge \forall z \, (Ry_0 z \to \neg \mathbf{P} z) \to \mathbf{Q} y \Big) \\ \wedge \forall y_1 \forall y_2 \Big(Rxy_1 \wedge Ry_1 y_2 \wedge \mathbf{Q} y_1 \to \neg \mathbf{P} y_2 \Big) \\ \to \exists z \Big(Rxz \wedge \forall w \, [Rzw \to \mathbf{P} w] \Big) \Big] \; .$$

- With the ordering $\mathbf{Q} < \mathbf{P}$ we have the form

$$\forall \mathbf{P} \forall \mathbf{Q} \forall x \quad \Big[\underbrace{\forall y_0 \Big(Rxy_0 \wedge \forall z \, (Ry_0 z \to \neg \mathbf{P} z) \to \mathbf{Q} y \Big)}_{UNI} \\ \wedge \underbrace{\forall y_1 \forall y_2 \Big(Rxy_1 \wedge Ry_1 y_2 \wedge \mathbf{Q} y_1 \to \neg \mathbf{P} y_2 \Big)}_{NEG} \\ \to \underbrace{\exists z \Big(Rxz \wedge \forall w \, [Rzw \to \mathbf{P} w] \Big)}_{POS} \Big] \; ,$$

  where UNI is a universal boxed form for $\mathbf{Q}$ and NEG (resp. POS) is a formula which is negative (resp. positive) with respect to $\mathbf{Q}$.

  Hence we are free to proceed with Step 4b of the extended algorithm. However, if we had chosen the ordering $\mathbf{P} < \mathbf{Q}$ then we are not in a form where instantiating is possible, since $\mathbf{P}$ occurs positively in both the antecedent and consequent of the main implication.

  This may suggest that, even after eliminating $\mathbf{Q}$, the algorithm is incapable of eliminating $\mathbf{P}$. However, as in Example 4.2.1, we get a surprising simplification.

- In Step 4b, we read off the minimal instances of $\mathbf{Q}$ as $\sigma(\mathbf{Q}) = \lambda u. \; Rxu \wedge \forall z \, (Ruz \to \neg \mathbf{P} z)$.

- In Step 5b, we instantiate $\mathbf{Q}$ to give us

$$\forall x \forall \mathbf{P} \quad \Big[\top \wedge \forall y_1 \forall y_2 \Big(Rxy_1 \wedge Ry_1y_2 \wedge Rxy_1 \wedge \forall z\,(Ry_1z \to \neg\mathbf{P}z) \to \neg\mathbf{P}y_2\Big) \to \exists z\Big(Rxz \wedge \forall w\,[Rzw \to \mathbf{P}w]\Big)\Big] \ .$$

  This antecedent is now tautologous; for suppose $\mathscr{N}$ is an $\mathscr{L}_2(R)$ such that

$$\mathscr{N} \models Rxy_1 \wedge Ry_1y_2 \wedge Rxy_1 \wedge \forall z\,(Ry_1z \to \neg\mathbf{P}z) \ .$$

  Then there are elements $w, v_1, v_2 \in W$ such that $(w, v_1) \in \widehat{R}$, and $(v_1, v_2) \in \widehat{R}$ and, whenever $u \in W$ is an $\widehat{R}$-successor of v_1, we have that $u \notin V(\mathbf{P})$.

  However, v_2 is an $\widehat{R}$-successor of v_1 so must have that $v_2 \notin V(\mathbf{P})$ and, consequently, $\mathscr{N} \models \neg\mathbf{P}y_2$. We have, then, that

$$\mathscr{N} \models \Big(Rxy_1 \wedge Ry_1y_2 \wedge Rxy_1 \wedge \forall z\,(Ry_1z \to \neg\mathbf{P}z) \to \neg\mathbf{P}y_2\Big) \ ,$$

  for all models $\mathscr{N}$, which is to say that it is a tautology.

- In Step 6b, we observe that the result of this instantiation is the formula

$$\forall x \forall \mathbf{P} \ \exists z\Big(Rxz \wedge \forall w\,[Rzw \to \mathbf{P}w]\Big) \ ,$$

  which remains in the form required by Step 4b of the extended algorithm, so we may repeat without issue.

- In Step 4b, this time around, the antecedent is empty, so we can read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.\ u \not\approx u$.

- Now in Step 5b, we apply the instantiation to give us the first-order formula $\forall x \exists z\,(Rxz \wedge \forall w \neg Rzw)$; unsurprisingly, the same property given by the SOQE-calculus.

Finally, we show in detail, the correctness of the instantiations used in the extended version of the Sahlqvist-van Benthem method.

Proposition 6.3.1. *The correctness of the instantiation used in the extended Sahlqvist-van Benthem algorithm can be deduced from the correctness of Nonnengart and Szałas' version of Ackermann's Lemma.*

Proof: This follows the same form as the proof of Proposition 4.3.1. However, the difference arises from our use of universal boxed forms instead of boxed formulas.

Suppose we have completed Steps 1-4 of the algorithm and have our Sahlqvist implication Φ in the form

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \Big[(REL \wedge UNI) \rightarrow POS^* \Big] \ ,$$

where UNI is a conjunction of translations of universal boxed forms, REL a conjunction of relational formulas and POS^* is positive. And suppose, without loss of generality, that $\mathbf{P}_1$ is the minimal second-order variable to be eliminated.

We then construct the set

$$\Sigma_i = \left\{ \tau(u) \ \middle| \ \text{ if } \forall u[\tau(u) \rightarrow \mathbf{P}_i u] \text{ is the PIA-form for some conjunct of } UNI \right\} .$$

Using the rules of Table 4.10, we can replace each conjunct of UNI with a semantically equivalent PIA-condition, so that we have Φ in the form

$$\forall x_1 \ldots \forall x_m \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \left[(REL \wedge PIA) \rightarrow POS^* \right] \ ,$$

where PIA is a conjunction of PIA-conditions. However, in this case our PIA-conditions may contain positive occurrences of second-order variables.

As before, we can assume without loss of generality, that there is precisely one PIA-condition with $\mathbf{P}_i$ occurring as the head. For convenience, let us write PIA_i for the unique PIA-condition with $\mathbf{P}_i$ occurring as the head and $\widehat{PIA_i}$ for the conjunction $\bigwedge_{j \neq i} PIA_j$.

Since $\mathbf{P}_1$ is the minimal second-order variable, we focus on the conjunct

PIA_1 in the subformula given below.

$$\forall \mathbf{P}_i \left[\forall u \left(\bigvee_{j=1}^{k} \tau_{i,j}(\vec{\mathbf{P}}, u) \rightarrow \mathbf{P}_1 u \right) \wedge REL \wedge \widehat{PIA_1} \rightarrow POS^* \right]$$

We negate this formula to give us,

$$\exists \mathbf{P}_i \left[\forall u \left(\bigvee_{j=1}^{k} \tau_{i,j}(\vec{\mathbf{P}}, u) \rightarrow \mathbf{P}_1 u \right) \wedge REL \wedge \widehat{PIA_1} \wedge \neg POS^* \right] \ ,$$

where we note that, in this case, $\mathbf{P}_1$ occurs only *negatively* in $\widehat{PIA_1}$ and, hence, we still have that $REL \wedge \widehat{PIA_1} \wedge \neg POS^*$ is negative with respect to $\mathbf{P}_1$. In this form, we are unable to apply Ackermann's Lemma, since the tail of PIA_1 may contain occurrences of the second-order variable $\mathbf{P}_1$. However, any such occurrences are positive and, consequently, we are able to apply Nonnengart and Szałas' version of Ackermann's Lemma. The substitution in this case being $\left\{ {}^{\mathbf{P}_1 u} / _{[\mathrm{LFP}\ \mathbf{P}_1.\ \bigvee_{j=1}^{k} \tau_{i,j}](u)} \right\}$.
Again, this is precisely the minimal instantiation given by the extended Sahlqvist-van Benthem method.

After we have instantiated $\mathbf{P}_1$, positive occurrences of $\mathbf{P}_1$ in the PIA-conditions of $\widehat{PIA_1}$ are replaced with fixed-point constructs which are positive in the variables $\mathbf{P}_2 \ldots \mathbf{P}_n$. As such, the instantiations do not force us outside the class of PIA-conditions, and, in particular, the monotonicity, which we were really interested in, is guaranteed by Proposition 2.3.1. Hence we may continue repeat this process for each of the remaining second-order variables. ∎

6.4 Summary

With these extended algorithms, the state of affairs is summed up by Table 6.3.

	*MA**	*SOQE**	*S-vB* method*
Sahlqvist class	✓	✓	✓
generalized Sahlqvist class	✓	✓	✓
simply generalized Sahlqvist class	✓	✓	✓

Table 6.3: Table indicating the success (✓) or failure (✗) of our extended algorithms for various classes.

All three algorithms are now well equipped for computing the correspondence properties of all modal axioms in the class of simply generalized Sahlqvist formulas. This class strictly extends to class of Sahlqvist formulas and the class of generalized formulas.

We note that in order to achieve this goal, we had to widen our search for correspondence properties to the first-order logic of fixed-points since there are formulas, Löb's axiom, which are simply generalized Sahlqvist formulas but have no first-order correspondence property.

Now, all that is lacking is the applicability of the MA-calculus to formulas of $\mathscr{L}(\Diamond, D)$. This is remedied with the introduction of the following rules:

D-Surfacing

$$\frac{N \cup \{\alpha \vee \overline{D}\beta(p)\}}{N \cup \{\overline{D}\alpha \vee \beta(p)}$$

D-Skolemization

$$\frac{N \cup \{\neg a \vee D\beta(p)\}}{N \cup \{\neg a \vee Db, \neg b \vee \beta(p)\}}$$

Table 6.4: Further rules for the MA-calculus

We note that these rules also preserve global satisfiability. That is to say:

Proposition 6.4.1. *The $\mathscr{L}_n^{\smile}(\Diamond)$-formula $\mathscr{M} \models \alpha \vee \overline{D}\beta(p)$ is globally satisfiable if and only if $\mathscr{M} \models \overline{D}\alpha \vee \beta(p)$ is globally satisfiable.*

Proof: Suppose that $\alpha \vee \overline{D}\beta(p)$ is globally satisfiable on some hybrid model $\mathscr{M}$, then for a fixed $w \in W$, we have $\mathscr{M}, w \models \alpha \vee \overline{D}\beta(p)$. We have three cases:

(i) The universe of $\mathscr{M}$ is the singleton set $\{w\}$ in which case it follows trivially that $\mathscr{M}, w \models \overline{D}\beta(p)$.

(ii) There is some $v \in W$, different to w, such that $\mathscr{M}, v \not\models \alpha$. In which case, since $\mathscr{M}, v \models \alpha \vee \overline{D}\beta(p)$ (by the definition of global satisfiability), we must have that $\mathscr{M}, v \models \overline{D}\beta(p)$. It then follows that $\mathscr{M}, w \models \beta(p)$.

(iii) Finally, for every $v \in W$ different to w, we have that $\mathscr{M}, v \models \alpha$. This is, then, the definition of $\mathscr{M}, w \models \overline{D}\alpha$.

In all cases, it follows that $\mathscr{M}, w \models \overline{D}\alpha \vee \beta(p)$. Since w was arbitrary, we have $\overline{D}\alpha \vee \beta(p)$ is globally satisfiable on $\mathscr{M}$. Due to the symmetric nature of the argument, the converse follows directly. ∎

Proposition 6.4.2. *The $\mathscr{L}_n^{\smile}(\Diamond, D)$-formula $\neg a \vee D\beta(p)$ is globally satisfiable if and only if $\bigwedge\{\neg a \vee Db, \neg b \vee \beta(p)\}$ if globally satisfiable.*

Proof: Suppose that $\neg a \vee D\beta(p)$ is globally satisfiable on some hybrid model $\mathscr{M} = (W, \widehat{R}, *, V)$, then we have that $\mathscr{M}, *(a) \models \neg a \vee D\beta(p)$ and, consequently, there is some $v \neq *(a)$ such that $\mathscr{M}, v \models \beta(p)$. Let b be a fresh nominal and let $\mathscr{M}'$ be the hybrid model $(W, \widehat{R}, \widetilde{*}, V)$, where

$$\widetilde{*}(c) = \begin{cases} *(c) & \text{if } c \neq b \\ v & \text{if } c = b \end{cases} .$$

Now, for a fixed $w \in W$, we have that $\mathscr{M}, w \models \neg a \vee D\beta(p)$. We have three cases:

(i) If $w \neq \widetilde{*}(a)$ and $w \neq \widetilde{*}(b)$ then we have that $\mathscr{M}', w \models \neg a$ and $\mathscr{M}', w \models \neg b$.

(ii) If $w = \widetilde{*}(a)$ and $w \neq \widetilde{*}(b)$ then $\mathscr{M}, w \models a$ and, by definition, we have $\mathscr{M}, w \models Db$. We also have that $\mathscr{M}', w \models \neg b$.

(iii) If $w \neq \widetilde{*}(a)$ and $w = \widetilde{*}(b)$ then $\mathscr{M}', w \models \neg a$ and we chose $\widetilde{*}(b)$ such that $\mathscr{M}, \widetilde{*}(b) \models \beta(p)$.

In all cases, we have that $\mathscr{M}', w \models \neg a \vee Db$ and $\mathscr{M}', w \models \neg b \vee \beta(p)$, and hence $\bigwedge\{\neg a \vee Db, \neg b \vee \beta(p)\}$ is globally satisfiable.

For the converse, suppose that $\mathscr{M}', w \models \bigwedge\{\neg a \vee Db, \neg b \vee \beta(p)\}$ for some $w \in W$, then it follows from the definitions that $\mathscr{M}', w \models \neg a \vee D\beta(p)$. Hence we have that if $\bigwedge\{\neg a \vee Db, \neg b \vee \beta(p)\}$ is globally satisfiable then $\neg a \vee D\beta(p)$ is globally satisfiable. ∎

For example:

Example 6.4.1. We show that the frame validity of the formula $(\neg p \wedge \overline{D}p) \to \Box\neg p$ corresponds to irreflexivity.

We apply the rules of this extended MA-calculus to the set $\{\neg a \vee (\neg p \wedge \overline{D}p \wedge \Diamond p)\}$, which corresponds to a refutation of $(\neg p \wedge \overline{D}p) \to \Box\neg p$:

~~1.~~	$\neg a \vee (\neg p \wedge \overline{D}p \wedge \Diamond p)$	
~~2.~~	$\neg a \vee \neg p$	Clausify, 1
~~3.~~	$\neg a \vee \overline{D}p$	Clausify, 1
~~4.~~	$\neg a \vee \Diamond p$	Clausify, 1
~~5.~~	$\overline{D}\neg a \vee p$	$\overline{D}$-Surfacing, 3
6.	$\neg a \vee \Diamond b$	Skolemization, 4
~~7.~~	$\neg b \vee p$	Skolemization, 4
8.	$\neg a \vee \neg(\neg \overline{D}\neg a \vee \neg b)$	Ackermann, 5,7 into 2

We now take the standard translation into first-order logic:

$$\Big[x \not\approx a \vee \exists y(Rxy \wedge y \approx b)\Big] \wedge \Big[x \not\approx a \vee \Big(\forall y(y \not\approx x \to y \not\approx a) \wedge x \approx b\Big)\Big]$$

This simplifies to Raa which we unskolemize to give us $\exists x Rxx$, which we negate to give us the first-order correspondence property $\forall x \neg Rxx$, as required.

With these extensions, all three algorithms have similar capabilities in computing correspondence properties. However, the algorithm which stands out as, perhaps, being least effective is the Sahlqvist-van Benthem method which is designed to act on modal formulas of a specific form; namely the class of Sahlqvist formulas (the extended algorithm being able to cope with the class of simply generalized Sahlqvist formulas). This is in contrast with the MA-calculus and the SOQE-calculus, which, although not always successful, can be applied to arbitrary modal formulas.

It would be interesting to see what bounds can be put on the computational powers of these two algorithms; can we describe a syntactic class for which the MA-calculus succeeds which is maximal (with respect to $\subseteq$)? Is this class strictly larger than the class of formulas that are semantically equivalent to simply generalized Sahlqvist formulas? And is it possible to further extend the Sahlqvist-van Benthem method so that it has the same computational capabilities as the MA-calculus and SOQE-calculus with regards to computing correspondence properties?

Chapter 7

Modal Logic with Propositional Quantifiers

7.1 Motivation

Former US Defense Secretary Donald Rumsfeld, when questioned about lack of evidence for weapons of mass destruction in Iraq, famously stated the following:

> *"As we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns; the ones we don't know we don't know."*

Figure 7.1: Secretary Rumsfeld at the Dept. of Defense news briefing in Feb 02 [1]

But what are we to infer from this in the context of epistemic logic? Clearly Rumsfeld's notion of knowledge is at odds with the traditional axiomatization of epistemic logic, which features the axiom $\mathbf{5} = \Diamond p \to \Box\Diamond p$, asserting that we are knowledgeable about the limitations of our knowledge. Clearly, if this is the case then the concept of "unknown unknowns" is ridiculous! However, it seems perfectly reasonable that we are not perfectly knowledgeable about what exactly we don't know. Did I know that I didn't know Einstein's theory of special relativity before I had ever been made aware of it? Of course, by their very nature, I cannot name an example of a current 'unknown unknown', for in naming it, I would have to *know* it was unknown. Debate over whether such things are 'unknown unknown' really exist should be placed on hold while we properly formalize the notion.

Suppose we interpret $\Box p$ as *'p is known'* then we can express the following:

R1$^-$	$\Box p \wedge \Box\Box p$	p is a known known,
R2$^-$	$\neg\Box p \wedge \Box\neg\Box p$	p is a known unknown,
R3$^-$	$\neg\Box p \wedge \neg\Box\neg\Box p$	p is an unknown unknown.

Perhaps, then, we should abandon axiom **5** as a legitimate representation of epistemic logic and introduce **R1$^-$**, **R2$^-$** and **R3$^-$** as axioms in its place. However, this is doomed to failure; modal axioms are implicitly universally quantified, which is far from what we intend, for we are not claiming that everything is an unknown unknown, only that there *exists* some things that are. To this end, we would like to be able to express formulas with explicit quantification over propositional variables. We could then express the following:

R1	$\exists p\ (\Box p \wedge \Box\Box p)$	There is some p which is a known known,
R2	$\exists p\ (\neg\Box p \wedge \Box\neg\Box p)$	There is some p which is a known unknown,
R3	$\exists p\ (\neg\Box p \wedge \neg\Box\neg\Box p)$	There is some p which is an unknown unknown.

But of course we need to formalize what we actually mean when we permit quantifiers in our language.

7.2 The language $\mathscr{L}^{\exists}(\Diamond)$

Modal logics with many-typed propositional quantifiers have been considered in the works of Fine [12], Kaplan [17] and Engelhardt et al. [11]. Here we restrict our focus to modal logics with propositional quantifiers of a single type.

Definition 7.2.1. Let the language of $\mathscr{L}^{\exists}(\Diamond)$ consist of a countable collection of propositional variables, a unique propositional constant $\bot$ together with the symbols $\neg$ (negation), $\wedge$ (conjunction), $\Diamond$ and $\exists$ as well as parentheses. Modal formulas of $\mathscr{L}^{\exists}(\Diamond)$ are built up according to the rule:

$$\varphi ::= p \ \mid \bot \mid \ \neg\varphi \ \mid \ (\varphi_1 \wedge \varphi_2) \ \mid \ \Diamond\varphi \ \mid \ (\exists p\ \varphi)\ ,$$

where p is some propositional variable.

We make use of the abbreviations in Table 3.1, together with the following abbreviation in Table 7.1.

name	*abbreviation*	*equivalent formula*
universal modality	$(\forall p\ \varphi)$	$\neg(\exists p\ \neg\varphi)$

Table 7.1: Aditional abbreviation for $\mathscr{L}^{\exists}(\Diamond)$.

We say that a propositional variable p occurs *free* in a formula φ if every occurrence of p occurs within the scope of the quantifier $\exists p$. We say that a modal formula φ is a *modal sentence* if no propositional variable occurs free in φ.

As with the basic modal language, truth of a modal formula of $\mathscr{L}^{\exists}(\Diamond)$ is defined over Kripke models $\mathscr{M} = (\mathscr{F}, V)$ where $\mathscr{F}$ is a Kripke frame and V is a valuation on $\mathscr{F}$.

We also recall the definition of a p-equivalent valuation, $V\left(\begin{smallmatrix} p \\ a \end{smallmatrix}\right)$, for each valuation V, which differs from V only at p, where it takes the value $a \subseteq W$. i.e.

$$V\left(\begin{smallmatrix} p \\ a \end{smallmatrix}\right)(q) = \begin{cases} V(q) & \text{if } p \neq q \\ a & \text{if } p = q \end{cases} .$$

For a given a model $\mathscr{M} = (\mathscr{F}, V)$ and p-equivalent valuation $V\left(\begin{smallmatrix} p \\ a \end{smallmatrix}\right)$, we construct the p-equivalent model, $\mathscr{M}\left(\begin{smallmatrix} p \\ a \end{smallmatrix}\right)$, with the same underlying Kripke frame $\mathscr{F}$ but with the valuation $V\left(\begin{smallmatrix} p \\ a \end{smallmatrix}\right)$.

We can, then, define truth of some modal formula φ in a Kripke model $\mathscr{M} = (W, \widehat{R}, V)$ inductively as follows. For any $w \in W$ we define:

$$\begin{aligned}
\mathscr{M}, w \models p &\iff w \in V(p) \text{ if } p \text{ is a propositional variable,} \\
\mathscr{M}, w \models \neg\varphi &\iff \mathscr{M}, w \not\models \varphi\ , \\
\mathscr{M}, w \models (\varphi \wedge \psi) &\iff \mathscr{M}, w \models \varphi \text{ and } \mathscr{M}, w \models \psi\ , \\
\mathscr{M}, w \models \Diamond\varphi &\iff (w, v) \in \widehat{R} \text{ and } \mathscr{M}, v \models \varphi \text{ for some } v \in W, \\
\mathscr{M}, w \models (\exists p\ \varphi) &\iff \mathscr{M}', w \models \varphi \text{ for some } p\text{-equivalent model } \mathscr{M}'.
\end{aligned}$$

We say that a modal formula φ is satisfiable if there is some Kripke model $\mathscr{M} = (W, \widehat{R}, V)$ and some $w \in W$ such that $\mathscr{M}, w \models \varphi$, and that φ is globally satisfiable if there is a model $\mathscr{M} = (W, \widehat{R}, V)$ such that $\mathscr{M}, w \models \varphi$ for all $w \in W$; we write $\mathscr{M} \models \varphi$. We say that φ is valid on a frame $\mathscr{F}$ if $\mathscr{M} \models \varphi$ for all models $\mathscr{M}$ with underlying frame $\mathscr{F}$; we write $\mathscr{F} \models \varphi$. Furthermore, we say that φ is valid on a class of frames Λ if $\mathscr{F} \models \varphi$ for all $\mathscr{F} \in \Lambda$.

Again, this semantics has a very natural translation into second-order logic. We extend the standard translation, π, defined in Chapter 3, to formulas of $\mathscr{L}^{\exists}(\Diamond)$ by providing the rule:

$$\pi\Big((\exists p \ \varphi), x\Big) = \Big(\exists \mathbf{P} \ \pi(\varphi, x)\Big) ,$$

where p is a propositional variable, φ is a modal formula and $\mathbf{P}$ is the second-order variable assigned to p.

We have, then, that for a given Kripke model $\mathscr{M} = (W, \widehat{R}, V)$ and any formula φ of $\mathscr{L}^{\exists}(\Diamond)$, $\mathscr{M}, w \models \varphi$ if and only if $\mathscr{N} \models \pi(\varphi, x)$, where $\mathscr{N} = (W', *, V')$ is an $\mathscr{L}_2(R)$-model with $W' = W$ and $*(R) = \widehat{R}$, where V' is any valuation such that $V'(x) = w$.

Given the semantics, as described above, we have the following lemma:

Lemma 7.2.1. *If φ is a sentence of $\mathscr{L}^{\exists}(\Diamond)$, that is, with no propositional variables occurring free, then φ is valid on a frame $\mathscr{F}$ if and only if φ is globally satisfiable on $\mathscr{F}$.*

Proof: If φ is valid on $\mathscr{F}$ then, by definition, φ is gloablly satisfiable on any model with underlying frame $\mathscr{F}$.

Conversely, suppose that φ is globally satisfiable on $\mathscr{F}$. That is to say that φ is gloablly satisfied in some model $\mathscr{M}$ whose underlying frame is $\mathscr{F}$. Taking the standard translation, we have that $\mathscr{N} \models \forall x \ \pi(\varphi, x)$, where $\mathscr{N}$ is the second-order model described above. Since no propositional variables occur free in φ and, consequently, no second-order variables occur free in $\pi(\varphi, x)$, it follows that we have $\mathscr{N} \models \forall \mathbf{P}_1 \ldots \forall \mathbf{P}_n \forall x \ \pi(\varphi, x)$. This is exactly the standard translation of the validity of φ on the frame $\mathscr{F}$, as required. ∎

We define a class of formulas in this language that have first-order correspondence properties which can be effectively computed.

Definition 7.2.2. We define *Sahlqvist atoms* as sentences of the form $\exists p_1 \ldots \exists p_n \psi$, where ψ is a Sahlqvist antecedent with free variables $p_1, \ldots, p_n$.

Sahlqvist sentences of $\mathscr{L}^{\exists}(\Diamond)$ are built up from Sahlqvist atoms by freely applying negations, conjunctions, and diamonds $\Diamond$.

Theorem 7.2.1. *Every Sahlqvist sentence has a first-order correspondence property, which can be effectively computed.*

Proof: We prove that every Sahlqvist atom is equivalent to a first-order correspondence property, it then follows that every Sahlqvist sentence has a first-order correspondence property.

Suppose φ is a Sahlqvist atom of the form $\exists p_1 \ldots \exists p_n \psi$ where ψ is a Sahlqvist antecedent.

First we pull all disjunctions to the left using the rules in Table 7.2.

input formula	*equivalent formula*
$(\alpha \vee \beta) \wedge \gamma$	$(\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$
$\exists p\, (\varphi \vee \psi)$	$(\exists p\ \varphi) \vee (\exists p\ \psi)$

Table 7.2: Rules for distributing over disjunctions.

We are then left with a disjunction of Sahlqvist atom containing no disjunctions. Without loss of generality, then, it suffices to prove the case for atoms that contain no disjunctions.

Suppose $\exists p_1 \ldots \exists p_n \psi'$ is such a formula and take the standard translation of the globally satisfiability, since by Lemma 7.2.1 this is the same as the translation of frame validity: $\forall x \exists \mathbf{P}_1 \ldots \exists \mathbf{P}_n \pi(\psi', x)$.

We rename variables such that no quantifier binds the same variable and that no quantifier binds x and pull out any existential quantifier to the front using the rules in Table 7.3.

input formula	*equivalent formula*
$(\exists x_i\ \alpha(x_i)) \wedge \beta$	$\exists x_i\,[\alpha(x_i) \wedge \beta]$, where x_i does not occur free in β
$\exists \mathbf{P} \exists x_i\ \psi$	$\exists x_i \exists \mathbf{P}\ \psi$

Table 7.3: Rules for pulling out existential quantifiers.

The result of this transformation is a formula of the form

$$\forall x \exists x_1 \ldots \exists x_m \exists \mathbf{P}_1 \ldots \exists \mathbf{P}_n \,(REL \wedge BOX \wedge NEG)\ , \tag{7.1}$$

where REL is a conjunction of formulas of the form Rx_ix_j, BOX is a conjunction of standard translations of boxed atoms and NEG is a conjunction of negative formulas.

We construct the sets

$$\Sigma_i = \left\{ \tau(u) \;\middle|\; \text{ if } \forall u[\tau(u) \to \mathbf{P}_i u] \text{ is the PIA-form for some conjunct of } BOX \right\}\ ,$$

for each $1 \leq i \leq n$. so that the minimal instances of $\mathbf{P}_i$ are given by the substitution,

$$\sigma(\mathbf{P}_i) = \lambda u.\ \bigvee \Sigma_i\ .$$

Re-writing each boxed formula of BOX in PIA-form and collecting together all those PIA-conditions with the same head into a single PIA-condition using the rules of Table 4.10, we may write PIA_i for the unique PIA-condition featuring $\mathbf{P}_i$ as the head and $\widehat{PIA}_i$ for the conjunction $\bigwedge_{j \neq i} PIA_j$.

Focusing on the conjunct PIA_i, we have, then, that φ is of the form

$$\exists x_1 \ldots \exists x_m \exists \mathbf{P}_1 \ldots \exists \mathbf{P}_n \left(\forall u \left[\bigvee_{j=1}^{k} \tau_{i,j}(u) \to \mathbf{P}_i u \right] \wedge REL \wedge \widehat{PIA}_i \wedge NEG \right)\ ,$$

where $\left(REL \wedge \widehat{PIA}_i \wedge NEG \right)$ is negative with respect to $\mathbf{P}_i$.

We are now in a position to use Ackermann's Lemma and apply the substitutions $\left\{ {}^{\mathbf{P}_i}/_{\sigma(\mathbf{P}_i)} \right\}$ where $\sigma(\mathbf{P}_i) = \lambda u.\ \bigvee_{j=1}^{k} \Sigma_i$ to eliminate the second-order variable $\mathbf{P}_i$.

The result is a formula containing no second-order variables, as required. ∎

7.3 Correspondence Properties of Rumsfeld

Returning to the case of known knowns (et al.), we see that each of the proposed axioms **R1**, **R2** and **R3** are Sahlqvist sentences and, thus, have first-order correspondence properties that can be effectively computed using the method outlined in the proof of Theorem 7.2.1.

Proposition 7.3.1. *The frame validity of **R1** corresponds to* $\top$*.*

Proof: We take the standard translation of the globally satisfiability of **R1**, which, by Lemma 7.2.1 is the same as the frame validity:

$$\exists x \exists \mathbf{P} \left[\underbrace{\forall y \, (Rxy \to \mathbf{P}y) \wedge \forall z \, [Rxz \to \forall w \, (Rzw \to \mathbf{P}w)]}_{BOX} \right] ,$$

where we have ensured that no quantifier binds the same variable and no quantifier binds x. We note that we are already in the form of (7.1) so we rewrite the universal boxed forms as a single PIA-condition to give us

$$\exists x \exists \mathbf{P} \forall u \Big[Rxu \vee \exists z \, (Rxz \wedge Rzu) \to \mathbf{P}u \Big] .$$

Now we apply the substitution $\left\{ {}^{bfP}/_{\sigma(\mathbf{P})} \right\}$ where $\sigma(\mathbf{P}) = \lambda u.\ Rxu \vee \exists z \, (Rxz \wedge Rzu)$. The resulting formula reduces to $\top$. That is to say that the formula $\exists p \ \Box p \wedge \Box\Box p$ is valid in every frame. ∎

Proposition 7.3.2. *The frame validity of **R2** corresponds to the first-order property* $\forall x \exists y Rxy$*.*

Proof: We take the standard translation of the globally satisfiability of **R2**:

$$\forall x \exists \mathbf{P} \left[\underbrace{\neg \forall y \, (Rxy \to \mathbf{P}y) \wedge \forall z \Big[Rxz \to \neg \forall w \, (Rzw \to \mathbf{P}w) \Big]}_{NEG} \right] .$$

Again, we already have the form (7.1) required to instantiate.

The conjunction BOX, here, appears empty so we may introduce the tautologous expression $\forall u \, (u \not\approx u \to \mathbf{P}u)$ and then apply the substitution

$\{{}^{\mathbf{P}}/_{\sigma(\mathbf{P})}\}$ where $\sigma(\mathbf{P}) = \lambda u.\ u \not\approx u$ to give us

$$\forall x \exists \mathbf{P} \Big[\neg\forall y\,(Rxy \rightarrow y \not\approx y) \wedge \forall z \big[Rxz \rightarrow \neg\forall w\,(Rzw \rightarrow w \not\approx w) \big] \Big] .$$

This formula reduces to $\forall x \Big(\exists y Rxy \wedge \forall z\,[Rxz \rightarrow \exists w Rzw] \Big)$, which is semantically equivalent to $\forall x \exists y Rxy$. That $\forall x \Big(\exists y Rxy \wedge \forall z\,[Rxz \rightarrow \exists w Rzw] \Big)$ implies seriality is a trivial weakening. For the converse we demonstrate the contrapositive. Consider the negation $\exists x \Big(\forall y \neg Rxy \vee \exists z [Rxz \wedge \forall w \neg Rzw] \Big)$, which stipulates that there is some $w \in W$ that has no $\widehat{R}$-successors, or, has some $\widehat{R}$-sucessor which has no $\widehat{R}$-successors. Which is to say that $\widehat{R}$ is not serial. Hence the frame validity of **R2** corresponds to seriality. ∎

And lastly we consider **R3**:

Proposition 7.3.3. *The frame validity of* ***R3*** *corresponds to the first-order property*

$$\forall x \exists y \exists z\,(Rxy \wedge Rxz \wedge \neg Rzy)$$

Proof: Again, we take the standard translation of the globally satisfiability of **R3**:

$$\forall x \exists \mathbf{P} \Big[\neg\forall y\,(Rxy \rightarrow \mathbf{P}y) \wedge \exists z \big[Rxz \wedge \forall w\,(Rzw \rightarrow \mathbf{P}w) \big] \Big] .$$

We then pull out the existential quantifiers to get

$$\forall x \exists z \exists \mathbf{P} \left[\underbrace{Rxz}_{REL} \wedge \underbrace{\neg\forall y\,(Rxy \rightarrow \mathbf{P}y)}_{NEG} \wedge \underbrace{\forall w\,(Rzw \rightarrow \mathbf{P}w)}_{UNI} \right] .$$

We can now apply the substitution $\{{}^{\mathbf{P}}/_{\sigma(\mathbf{P})}\}$ where $\sigma(\mathbf{P}) = \lambda u.\ Rzu$ to give us

$$\forall x \exists z \Big[Rxz \wedge \neg\forall y\,(Rxy \rightarrow Rzy) \wedge \forall w\,(Rzw \rightarrow Rzw) \Big] ,$$

which reduces to $\forall x \exists y \exists z\,(Rxy \wedge Rxz \wedge \neg Rzy)$; i.e. non-euclideanness. ∎

Furthermore, we see that **R3** $\Rightarrow$ **R2** $\Rightarrow$ **R1** since the class of non-euclidean frames is subsumed by the class of serial frames which is, itself, subsumed by the class of all frames.

However, upon further inspection, **R1** and **R2** are not what Rumsfeld asserts; he claims that "[we] *know* there are known knowns [. . . and] we also *know* there are known unknowns". We might better wish to consider the following axioms **R1$^+$** and **R2$^+$** in place of **R1** and **R2**.

R1$^+$	$\Box(\exists p\ \Box p \wedge \Box\Box p)$	We known there is some p that is a known known,
R2$^+$	$\Box(\exists p\ \neg\Box p \wedge \Box\neg\Box p)$	We know there is some p that is a known unknown.

But these formulas are again Sahlqvist sentences of $\mathscr{L}^{\exists}(\Diamond)$ and so the same procedure will similarly apply. We find that **R1$^+$** corresponds to $\forall x \forall y\,(Rxy \rightarrow \top)$, which again reduces to $\top$, and that **R2$^+$** corresponds to $\forall x \forall v\,(Rxv \rightarrow \exists y Rvy)$. This is better expressed as $\forall v\,(\exists x Rxv \rightarrow \exists y Rvy)$, which says that for every $w \in W$, if w has an $\widehat{R}$-predecessor then w has an $\widehat{R}$-successor.

7.4 Comparing $\mathscr{L}^{\exists}(\Diamond)$ with $\mathscr{L}(\Diamond, D)$

One thing we notice is that **R3** corresponds to non-euclideanness, a property not definable in the basic modal language $\mathscr{L}(\Diamond)$. Other properties that are definable in $\mathscr{L}^{\exists}(\Diamond)$ but not in $\mathscr{L}(\Diamond)$ are given in Table 7.4, all of which are Sahlqvist atoms whose correspondence properties can be computed in the same manner as above.

Modal Formula	*Correspondence Property*	
$\exists p\,(p \wedge \Box\neg p)$	$\forall x \neg Rxx$	(Irreflexivity)
$\exists p\,(p \wedge \Box\Box\neg p)$	$\forall x \forall y\,(Rxy \rightarrow \neg Ryx)$	(Asymmetry)
$\exists p\,(\Box\neg p \wedge \Box\Box p)$	$\forall x \forall y \forall z\,(Rxy \wedge Rxz \rightarrow \neg Ryz)$	(Full Non-euclideanness)
$\exists p\,(\neg\Box p \wedge \Diamond\Box p)$	$\forall x \exists y \exists z\,(Rxy \wedge Rxz \rightarrow \neg Ryz)$	(Non-euclideanness)

Table 7.4: Correspondence properties in $\mathscr{L}^{\exists}(\Diamond)$.

For example:

Proposition 7.4.1. *The $\mathscr{L}^{\exists}(\Diamond)$-sentence $\exists p \ (p \wedge \Box\neg p)$ corresponds to irreflexivity.*

Proof: By Lemma 7.2.1, frame validity is equivalent to global satisfiability for modal sentences so we take the standard translation of the globally satisfiability of $\exists p \ (p \wedge \Box\neg p)$:

$$\forall x \exists \mathbf{P} \ \left[\underbrace{Px}_{UNI} \wedge \underbrace{\forall y \, (Rxy \rightarrow \neg \mathbf{P} y)}_{NEG} \right] .$$

We read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u. \ u \approx x$ and substitute to give us

$$\forall x \Big(x \approx x \wedge \forall y \, [Rxy \rightarrow \neg \, (x \approx y)] \Big) ,$$

which reduces to $\forall x \neg Rxx$, as required ∎

As well as some correspondence properties not definable in $\mathscr{L}(\Diamond)$, we can also define some modal operators that are not definable in $\mathscr{L}(\Diamond)$; such as the graded modalities $(\geq n)$, $(\leq n)$ and $(= n)$, which we discussed in Chapter 3.

Proposition 7.4.2. *The formula $(\geq n)\varphi$ is definable in $\mathscr{L}^{\exists}(\Diamond)$ by the formula*

$$\exists p_1 \ldots \exists p_n \left[\Diamond \, (p_1 \wedge \varphi) \wedge \ldots \wedge \Diamond \, (p_n \wedge \varphi) \wedge \bigwedge_{1 \leq i < j \leq n} \Box \, (\neg p_i \vee \neg p_j) \right] . \tag{7.2}$$

Proof: We use the same approach as above for eliminating the existential quantifiers. The resulting formula will take the same form as the standard translation of $(\geq n)\varphi$.

Taking the standard translation of (7.2), evaluated at x, we have

$$\exists \mathbf{P}_1 \ldots \exists \mathbf{P}_n \Bigg[\exists y_1 \Big(Rxy_1 \wedge \mathbf{P}_1 y_1 \wedge \pi(\varphi, y_1) \Big) \wedge \ldots \wedge \exists y_n \Big(Rxy_n \wedge \mathbf{P}_n y_n \wedge \pi(\varphi, y_n) \Big)$$

$$\wedge \bigwedge_{i \neq j} \forall z \, (Rxz \rightarrow \neg \mathbf{P}_i z \vee \neg \mathbf{P}_j z) \Bigg] .$$

Pulling out the existential quantifiers we have:

$$\exists y_1 \ldots \exists y_n \exists \mathbf{P}_1 \ldots \exists \mathbf{P}_n \Bigg[\underbrace{Rxy_1 \ldots \wedge Rxy_n}_{REL} \wedge \underbrace{\mathbf{P}_1 y_1 \wedge \ldots \wedge \mathbf{P}_n y_n}_{UNI}$$

$$\underbrace{\wedge\, \pi(\varphi, y_1) \wedge \ldots \wedge \pi(\varphi, y_n) \wedge \bigwedge_{i \neq j} \forall z \, (Rxz \to \neg \mathbf{P}_i z \vee \neg \mathbf{P}_j z)}_{NEG} \Bigg] .$$

We now substitute $\mathbf{P}_i$ for $\sigma(\mathbf{P}_i) = \lambda u.\ u \approx y_i$ for each $1 \leq i \leq n$ to get:

$$\exists y_1 \ldots \exists y_n \Big[Rxy_1 \ldots \wedge Rxy_n \wedge y_1 \approx y_1 \wedge \ldots \wedge y_n \approx y_n$$

$$\wedge\, \pi(\varphi, y_1) \wedge \ldots \wedge \pi(\varphi, y_n) \wedge \bigwedge_{i \neq j} \forall z \Big(Rxz \to \neg\, (z \approx y_i) \vee \neg\, (z \approx y_j) \Big) \Big] .$$

Now each of the conjuncts of the form $\forall z \Big(Rxz \to \neg\, (z \approx y_i) \vee \neg\, (z \approx y_j) \Big)$ can be equivalently rewritten as $\forall z \Big(Rxz \wedge z \approx y_i \to \neg\, (z \approx y_j) \Big)$. The antecedent of this implication is satisfied only when $z = y_i$, in which case we deduce that $y_i \neq y_j$. From this we deduce that the n $\widehat{R}$-successors must be distinct. This is precisely the definition of $(\geq n)\varphi$, evaluated at x. ∎

Alternative formulations of graded modalities in this language are given in [17] and [11] but the quantifier elimination required for demonstrating their equivalence is less straightforward.

It is interesting to note that although the properties irreflexivity, asymmetry and full non-euclideanness given in Table 7.4 are not definable in the basic modal language $\mathscr{L}(\Diamond)$, they *are* definable in the extended language $\mathscr{L}(\Diamond, D)$, by making use of the difference operator [8].

We see, though, that despite their similarities, the expressive power of $\mathscr{L}^{\exists}(\Diamond)$ is, in fact, greater than that of $\mathscr{L}(\Diamond, D)$.

We showed that the graded modalities $(\geq n)$ are definable in $\mathscr{L}^{\exists}(\Diamond)$, however we showed in Proposition 3.5.3 that these are not definable in $\mathscr{L}(\Diamond, D)$, since their satisfiability is not invariant under $\neq$-bisimulations. Hence, there are formulas expressible in $\mathscr{L}^{\exists}(\Diamond)$, for which there can be no semantic equivalent expression in $\mathscr{L}(\Diamond, D)$.

Leaving aside, for now, the question of whether there are also formulas expressible in $\mathscr{L}(\Diamond, D)$ for which there is no equivalent expression in $\mathscr{L}^{\exists}(\Diamond)$, we surely must have that $\mathscr{L}^{\exists}(\Diamond, D)$ - the extension of $\mathscr{L}(\Diamond, D)$ by propositional quantifiers - shares at least the combined expressivity of both languages and, therefore, strictly extends the expressivity of $\mathscr{L}(\Diamond, D)$.

Proposition 7.4.3. *The formula $\Diamond^{\smile}\varphi$ is definable in $\mathscr{L}^{\exists}(\Diamond, D)$ by the formula*

$$\exists p \ \Big(p \wedge \overline{D}\neg p \wedge \Diamond^{\smile}(\varphi \wedge \Diamond p)\Big) \ . \tag{7.3}$$

Proof: Taking the standard translation of (7.3), evaluated at x, we have:

$$\exists \mathbf{P}\Big(\mathbf{P}x \wedge \forall y[y \not\approx x \to \neg \mathbf{P}y] \wedge \exists z\,[\pi(\varphi, z) \wedge \exists w(Rzw \wedge \mathbf{P}w)]\Big) \ .$$

We now pull out the existential quantifiers to get a form where we can apply Ackermann's Lemma; namely

$$\exists z \exists w \exists \mathbf{P}\Big(\underbrace{Rzw}_{REL} \wedge \underbrace{\mathbf{P}x \wedge \mathbf{P}w}_{UNI} \wedge \underbrace{\forall y[y \not\approx x \to \neg \mathbf{P}y] \wedge \pi(\varphi, z)}_{NEG}\Big) \ .$$

We now substitute $\mathbf{P}$ with $\sigma(\mathbf{P}) = \lambda u.(u \approx x) \vee (u \approx w)$, which results in

$$\exists z \exists w\Big(Rzw \wedge \forall y[y \not\approx x \to (y \not\approx x \wedge y \not\approx w)] \wedge \pi(\varphi, z)\Big) \ ,$$

which reduces to $\exists z\Big(Rzx \wedge \pi(\varphi, z)\Big)$. This is exactly the standard translation of $\Diamond^{\smile}\varphi$, evaluated at x, as required. ∎

However, we should be cautionary of ever increasing expressivity in this way, for we see that the domino problem is now reducible to the satisfiability of some $\mathscr{L}^{\exists}(\Diamond, D)$-formula. Hence, we have inadvertently introduced undecidability into our language.

Theorem 7.4.1. *The satisfiability problem for $\mathscr{L}^{\exists}(\Diamond, D)$ is undecidable.*

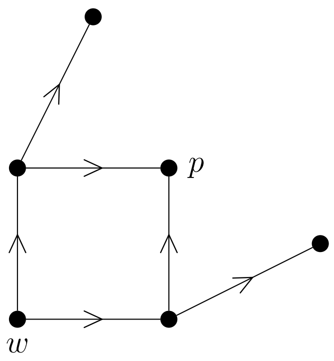
Proof: We first need to formulate an expression φ_{grid}, which forces all models of φ_{grid} to have a grid-like structure. This is achievable with the formula

$$\varphi_{grid} = (=2)\top \wedge \exists p(p \wedge \Box\Box\neg p) \wedge \exists p\Big[\Box\Diamond p \wedge \forall q\Big(\Diamond\Box q \to \Box\Box(p \to q)\Big)\Big] .$$

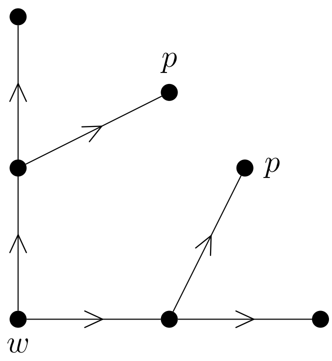
This formidable looking formula is worth discussing. Suppose that $\mathscr{M} = (W, \widehat{R}, V)$ is a model that satisfies φ_{grid} at $w \in W$, then $\mathscr{M}, w \models (=2)\top$; which is to say that w has exactly 2 R-successors. We have, also, that $\mathscr{M}, w \models \exists p(p \wedge \Box\Box\neg p)$, which we encountered earlier in this chapter. This stipulates that the accessibility relation is locally asymmetric at w. Finally, we have that

$$\mathscr{M}, w \models \exists p\Big[\Box\Diamond p \wedge \forall q\Big(\Diamond\Box q \to \Box\Box(p \to q)\Big)\Big] ,$$

which says, firstly, that there is some p that is accessible from both of w's $\widehat{R}$-successors. Hence we must be in either Case 1 or Case 2 below.

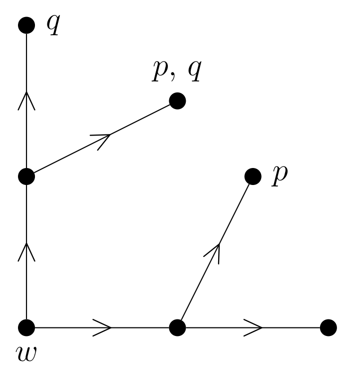


Case 1



Case 2

It also stipulates that for all valuations of q, if $\Box q$ is satisfied by one of w's $\widehat{R}$-successors, then $\Box q$ is in fact satisfied by both $\widehat{R}$-successors. This forces all models of φ_{grid} to have the form of Case 1, since Case 2 as the following diagram illustrates.



Motivation for this formula comes from the axiom $\mathbf{AT} = \exists p\Big(p \wedge \forall q[q \rightarrow \Box(p \rightarrow q)]\Big)$ of [17], which, together with the **S5** axioms, stipulate that there is a propositional variable which is true only at the current world. And although very different in nature, some similarities in the syntactic structure can be noted.

We now need a means of talking about horizontal $\widehat{R}$-successors and vertical $\widehat{R}$-successors with a single relation. To do this, we enforce a three-colour checker-board pattern (see Figure 7.2) upon the universe. We define

$$
\begin{aligned}
\psi_1 &= \Big(b \wedge \neg w \wedge \neg g\Big) \vee \Big(\neg b \wedge w \wedge \neg g\Big) \vee \Big(\neg b \wedge \neg w \wedge g\Big)\ , \\
\psi_2 &= b \rightarrow \Big(\Diamond w \wedge \Diamond g\Big)\ , \\
\psi_3 &= w \rightarrow \Big(\Diamond g \wedge \Diamond b\Big)\ , \\
\psi_4 &= g \rightarrow \Big(\Diamond b \wedge \Diamond w\Big)\ ,
\end{aligned}
$$

then let $\varphi_{check} = \boxed{\forall}\varphi_{grid} \wedge \boxed{\forall}\psi_1 \wedge \boxed{\forall}\psi_2 \wedge \boxed{\forall}\psi_3 \wedge \boxed{\forall}\psi_4$. We have, then, that $\mathscr{M} \models \varphi_{check}$ if and only if the valuations $V(b), V(g), V(w)$ form a partition of the universe that matches the three-colour checker-board pattern depicted below.

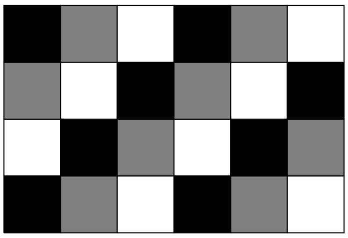


Figure 7.2: Illustation of a three-colour checker-board pattern.

Using the checker-board colours, we allow ourselves consistency when talking about horizontal and vertical $\widehat{R}$-successors. Let us use the following abbreviations

$$
\begin{aligned}
Right(\varphi) &\iff \Big(b \to \Diamond(g \wedge \varphi)\Big) \vee \Big(g \to \Diamond(w \wedge \varphi)\Big) \vee \Big(w \to \Diamond(b \wedge \varphi)\Big) \;, \\
Up(\varphi) &\iff \Big(b \to \Diamond(w \wedge \varphi)\Big) \vee \Big(g \to \Diamond(b \wedge \varphi)\Big) \vee \Big(w \to \Diamond(g \wedge \varphi)\Big) \;.
\end{aligned}
$$

Now, up to reflections in the main diagonal, we have that $\mathscr{M}, w \models \mathrm{Right}(\varphi)$ if and only if $\mathscr{M}, v \models \varphi$, where v is the unique $\widehat{R}$-successor directly to the right of w. And that $\mathscr{M}, w \models \mathrm{Up}(\varphi)$ if and only if $\mathscr{M}, v \models \varphi$, where v is the unique $\widehat{R}$-successor directly above w. Where $\mathscr{M}$ is any model satisfying φ_{check}.

We are now in a position to describe the reduction of the domino problem to the satisfiability problem of $\mathscr{L}^{\exists}(\Diamond)$. Given a domino system $\mathcal{D} = (T, H, V)$, where $T = \{D_1, \ldots, D_n\}$ is a set of domino types and $H, V \subseteq T \times T$ are the horizontal and vertical conditions. We associate each D_i of T with an proposition variable d_i and refer, also, to the propositional variables b, w and g used above. We now construct the following set of $\mathscr{L}^{\exists}(\Diamond, D)$-formulas:

$$
\begin{aligned}
\psi_1 &= d_1 \vee \ldots \vee d_n \;, \\
\psi_2 &= \bigwedge_{i=1}^{n-1} \left(d_i \to \bigwedge_{j=i}^{n-1} \neg d_{j+1} \right) \;, \\
\chi &= \bigwedge_{i=1}^{n} \left[d_i \to \bigvee_{(D_i, D_j) \in H} Right(d_j) \;\wedge \bigvee_{(D_i, D_k) \in V} Up(d_k) \right] \;.
\end{aligned}
$$

We have that the formula $\Phi = \varphi_{check} \wedge \boxed{\forall}(\psi_1 \wedge \psi_2 \wedge \chi)$ is satisfiable if and only if the domino system $\mathcal{D}$ has a tiling.

Suppose that $\mathscr{M}$ is a model and $w \in W$ such that $\mathscr{M}, w \models \Phi$, then, by unfolding the definition of the global modality $\boxed{\forall}$, we have that $\mathscr{M}, v \models \left(\varphi_{check} \wedge \psi_1 \wedge \psi_2 \wedge \chi\right)$ for all $v \in W$. In particular, $\mathscr{M} \models \varphi_{check}$, from which we can deduce that there is some function $f : \mathbb{N} \times \mathbb{N} \to W$ such that $f(0,0) = w$ and $\Big(f(n,m), f(n{+}1,m)\Big) \in R$ and $\Big(f(n,m), f(n{+}1,m)\Big) \in R$ for all $n, m \in \mathbb{N}$. Furthermore, we have that $V(b), V(g), V(w)$ form a partition on the universe that conforms to the pattern given in Figure 7.2.

We define the tiling $t : \mathbb{N} \times \mathbb{N} \to T$ by

$$t(n,m) = d_i \Longleftrightarrow \mathscr{M}, f(n,m) \models d_i \ .$$

Since $\mathscr{M}, v \models \psi_1 \wedge \psi_2$ for all $v \in W$, we can be sure that this definition is total and well-defined. We also have that the tiling t must satisfy the necessary horizontal and vertical conditions since $\mathscr{M}, v \models \chi$ for all $v \in W$.

For the converse, suppose that $t : \mathbb{N}\times\mathbb{N} \to T$ is a tiling of $\mathcal{D}$, then we define the model $\mathscr{M} = (W, \widehat{R}, V)$ as follows:

$$\begin{aligned}
W &= \mathbb{N} \times \mathbb{N} \ , \\
\widehat{R} &= \Big\{ \Big((n,m), (n+1,m)\Big), \Big((n,m), (n,m+1)\Big) \ \mid \ n, m \in \mathbb{N} \Big\} \ , \\
V(b) &= \{(n,m) \ \mid \ n \equiv m \mod 3\} \ , \\
V(g) &= \{(n,m) \ \mid \ n \equiv m+1 \mod 3\} \ , \\
V(w) &= \{(n,m) \ \mid \ n \equiv m+2 \mod 3\} \ , \\
V(d_i) &= \{(n,m) \ \mid \ t(n,m) = D_i\} \ .
\end{aligned}$$

It is then easily checked that $\mathscr{M}, w \models \Phi$ for any $w \in W$.

Since the domino problem is undecidable [3], and is effectively reducible to a satisfiability problem in $\mathscr{L}^{\exists}(\Diamond)$, so too must the satisfiability problem of $\mathscr{L}^{\exists}(\Diamond, D)$ be undecidable. ∎

The extension to $\mathscr{L}^{\exists}(\Diamond, D)$ was slightly artificial, in order to allow us the availability of the global quantifier $\boxed{\forall}$, which is necessary for this proof. However, it is known that the language $\mathscr{L}^{\exists}(\Diamond)$ is already undecidable. Kaminski and Tiomkin [16], building upon the work of Fine [12], show that the expressive power of the basic $\mathscr{L}^{\exists}(\Diamond)$ is equivalent to that of second-order logic and, as such, the undecidability of $\mathscr{L}^{\exists}(\Diamond)$ follows from the undecidability of $\mathscr{L}_2(\tau)$.

To conclude, we consider the epistemic theory of $\mathscr{L}^{\exists}(\Diamond)$ given by the axioms in Table 7.5.

Name	*Formula*	*Epistemic Interpretation*
T	$\Box p \to p$	**True Knowledge**: Everything that is known to be true is actually true.
4	$\Box p \to \Box\Box p$	**Positive introspection**: Everything that is known, is known to be known. That is to say we have perfect knowledge of what we know.
R3	$\exists p\,(\Diamond p \land \Diamond\Box\neg p)$	**Unknown Unknowns**: There are things we don't know and are unaware of our ignorance of them.

Table 7.5: Rumsfeld's axioms for the epistemic interpretation of $\mathscr{L}^{\exists}(\Diamond)$.

As discussed earlier, the class of frames on which **R3** holds is subsumed by the class of frames on which **R2**$^+$ and **R1**$^+$ hold and, hence, both **R1**$^+$ and **R2**$^+$ are logical consequences of **R3** and so need not be included in our theory.

The class of frames that satisfy this theory is larger than the class of reflexive, transitive and convergent frames and, hence, by the result of [16], has the same expressive power as full second-order logic and is, therefore, undecidable. This stands in contrast with usual epistemic interpretation, where the modality is **S5**, which is decidable and has a complete axiomatic system.

There are, therefore, statements that are consistent with Rumsfeld's claims to the Dept of Defense news briefing (Figure 7.1) but the truth or falsity of which cannot be ascertained be any given decision procedure. So, whether we believe Mr Rumsfeld's remarks to be ridiculous or profound, one thing we can agree upon, is that they are a poor basis for a tractable theory of knowledge.

Chapter 8

Conclusion

We have discussed the basic modal language and several extensions involving the difference operator, global modalities, graded modalities and converse modalities. We showed, by an invariance result for bisimilar models in Chapter 3, that each of these extensions strictly increase the expressive power of the basic modal logic.

In Chapter 4, we explored three methods for computing first-order correspondence properties, which successfully terminate on a large class of modal formulas, namely the generalized Sahlqvist class. Two of these algorithms were based on Ackermann's Lemma, whereas the third involved identifying the minimal instances of second-order predicates and exploiting monotonicity. We saw how the success of the MA-calculus and the SOQE-calculus was largely dependent on the order in which we chose to eliminate second-order variables (see Examples 4.1.1 and 4.2.1). However, for the class of generalized Sahlqvist formulas, successful elimination was guaranteed for *any* ordering on the second-order variables.

We explored the relationships between these three algorithms in Chapter 5, demonstrating that the SOQE-calculus is, indeed, a second-order adaptation of the MA-calculus, in that every rule of the MA-calculus can be emulated by a finite sequence of rules of the SOQE-calculus. We also showed how the instantiation process of the Sahlqvist-van Benthem method can be seen as an application of Ackermann's Lemma. It was remarked that both the SOQE-calculus and the Sahlqvist-van Benthem method are not restricted to the basic modal language and are equally successful in computing correspondence properties for formulas of $\mathscr{L}(\Diamond, D)$ and $\mathscr{L}_n^{\smile}(\Diamond)$.

Having described the relative successes and failures of each of the three algorithms, we described some extensions which allowed each algorithm similar computational capabilities. In particular, we extended each algorithm to successfully compute correspondence properties for simply generalized Sahlqvist formulas, in the first-order logic of fixed-points.

In pursuing a modal langauge capable of formalizing Rumsfeld's notion of 'unknown unknowns' (et al.) in Chapter 7, we described the syntax and semantics for the modal logic with propositional quantifiers $\mathscr{L}^{\exists}(\Diamond)$. We showed that a simple class of $\mathscr{L}^{\exists}(\Diamond)$-formulas had first-order correspondence properties which, by a variation on the Sahlqvist-van Benthem, could be effectively computed. The class of Sahlqvist sentences we described, is far from a complete description of all formulas of this language that has first-order properties. Indeed, by the result of Chagrova [6], no such class can be given. However, there are sure to be some larger class of formulas with first-order correspondence properties whose syntactic definition is relatively simple. We did not pursue such definitions here. We concluded this chapter by remarking that, due to the undecidability of $\mathscr{L}^{\exists}(\Diamond)$, which we discussed, Rumsfeld's speech makes for a poor epistemic theory of knowledge.

Further area for discourse would have been axiomatics and proof theory for modal logic with propositional quantifiers, as is discussed in [17, 11], which describe completeness results for their respective proof theories. However, due to the otherwise model-theoretic nature of this dissertation, we, with reluctance, gave no discussion of this here.

Bibliography

[1] DoD news briefing - Secretary Rumsfeld and Gen. Myers, Feburary 2002. `http://www.defense.gov/transcripts/transcript.aspx?transcriptid=2636`.

[2] Wilhelm Ackermann. Untersuchungen über das Eliminationsproblem der Mathematischen Logik. *Mathematisce Annalen*, 110:390–413, 1935.

[3] Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematics Society*, 66, 1966.

[4] Nick Bezhanishvili and Ian Hodkinson. Sahlqvist theorem for modal fixed point logic. *submitted*, 2010.

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

[6] Lilia A. Chagrova. An undecidable problem in correspondence theory. *The Journal of Symbolic Logic*, 56:1261–1272, 1991.

[7] Willem Ernst Conradie. *Algorithmic Correspondence and Completeness in Modal Logic*. PhD thesis, School of Mathematics, University of Witwatersrand, 2007.

[8] Maarten de Rijke. The modal logic of inequality. *The Journal of Symbolic Logic*, pages 566–584, 1992.

[9] Maarten de Rijke. *Extending Modal Logic*. PhD thesis, Institute of Logic, Language and Computation, University van Amsterdam, 1993.

[10] Herbert B. Enderton. *An Introduction to Mathematical Logic*, chapter 4: Second-order Logics. Academic Press, 2nd edition, 2001.

[11] Kai Engelhardt, Ron van der Meyden, and Kaile Su. Modal logic with a linear heirarchy of propositional quantifiers. *In proceedings of Advanced in Modal Logic*, 4:9–30, 2003.

[12] Kit Fine. Propositional quantifiers in modal logic. *Theoria*, 36:336–346, 1970.

[13] Dov M. Gabbay, Renate A. Schmidt, and Andrzej Szałas. *Second-Order Quantifier Elimination.* College Publications, 2008.

[14] Valentin Goranko and Dimiter Vakarelov. *Annals of Pure and Applied Logic*, 141:180–217, 2006.

[15] Wilfred Hodges. *The Blackwell Guide to Philosophical Logic*, chapter 1: Classical Logic I: First-Order Logic, pages 9–31. Blackwell Publishing, 2001.

[16] Michael Kaminski and Michael Tiomkin. The expressive power of second-order propositional modal logic. *Notre Dame Journal of Formal Logic*, 37, 1996.

[17] David Kaplan. S5 with quantifiable propositional variables. *The Journal of Symbolic Logic*, 35:355, 1970.

[18] William W. McCune. Un-skolemizing clause sets. *Information Processing Letters*, 29, 1988.

[19] Elliott Mendelson. *An Introduction to Mathematical Logic*, chapter Appendix: Second-order Logic. Chapman *&* Hall, 4th edition, 1997.

[20] Andreas Nonnengart and Andrzej Szałas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. *Studies in Fuzziness and Soft Computing*, 24:307–328, 1998.

[21] Henrik Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. *Studies in Logic and the Foundations of Mathematics*, 82:110–143, 1975.

[22] Renate Schmidt. Modal logics and description logics: The SOQE calculus. *Unpublished course materials produced for The University of Manchester*, 2011.

[23] Renate Schmidt. The Ackermann approach for modal logic, correspondence theory and second-order reduction. *To appear in Journal of Applied Logic*, 2011.

[24] Johan van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1985.

[25] Johan van Benthem. Modal correspondences and fixed-points. *Studia Logica: An International Journal for Symbolic Logic*, 83:133–155, 2006.

[26] Johan van Benthem and Kees Doets. *Handbook of Philisophical Logic*, chapter 3: Higher-order Logics. Springer, 2nd edition, 2001.

[27] Hao Wang. Proving theorems by pattern recognition II. *Bell System Tech. Journal*, 40:1–41, 1961.

Appendix A

Further Correspondence Examples

For completeness, we offer the proofs of the correspondence properties stated in Tables 3.6 and 7.4. The correspondence properties given in Table 3.3 are standard and so we will not detail their proofs here.

Example A.0.1. The formula $(\neg p \wedge \overline{D}p) \to \Box p$ corresponds to irreflexivity.

We take the standard translation of the frame validity of $(\neg p \wedge \overline{D}p) \to \Box p$, that is with all free variables universally quantified:

$$\forall x \forall \mathbf{P} \left[\underbrace{\neg \mathbf{P} x}_{NEG} \wedge \underbrace{\forall y (x \not\approx y \to \mathbf{P} y)}_{UNI} \to \underbrace{\forall z (Rxz \to \mathbf{P} z)}_{POS} \right] .$$

We move NEG across to the consequent of the main implication and read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.\ x \not\approx u$. Applying the minimal instantiation, we get

$$\forall x \left[\forall y (x \not\approx y \to x \not\approx y) \to \Big(x \not\approx x \vee \forall z (Rxz \to x \not\approx z) \Big) \right] ,$$

which reduces to $\forall x \neg Rxx$, as required.

Example A.0.2. The formula $\Big(p \wedge \Diamond(\Diamond p \wedge \neg p)\Big) \to \overline{D}p$ (adapted from the semantically equivalent $(p \wedge \overline{D}\neg p) \to \Box(\Diamond p \to p)$ of [8]) corresponds to anti-symmetry.

We take the standard translation of the frame validity of $\Big(p \wedge \Diamond(\Diamond p \wedge \neg p)\Big) \to \overline{D}p$:

$$\forall x \forall \mathbf{P}\left[\mathbf{P}x \wedge \exists y\Big(Rxy \wedge \exists z[Ryz \wedge \mathbf{P}z] \wedge \neg \mathbf{P}y\Big) \to \exists w(x \not\approx w \to \mathbf{P}w)\right].$$

We pull out all existential quantifiers from the antecedent to get

$$\forall x \forall y \forall z \forall \mathbf{P}\left[\underbrace{Rxy \wedge Ryz}_{REL} \wedge \underbrace{\mathbf{P}x \wedge \mathbf{P}z}_{UNI} \wedge \underbrace{\neg \mathbf{P}y}_{NEG} \to \underbrace{\exists w(x \not\approx w \to \mathbf{P}w)}_{POS}\right],$$

and moving NEG across the main implication, we can read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.(u \approx x) \vee (u \approx z)$.

Applying the minimal instantiation, we get

$$\forall x \forall y \forall z\left[Rxy \wedge Ryz \to \Big[(y \approx x) \vee (y \approx z)\Big] \vee \exists w\Big(x \not\approx w \to \Big[(w \approx x) \vee (w \approx z)\Big]\Big)\right].$$

The subformula $\exists w\Big(x \not\approx w \to \Big[(w \approx x) \vee (w \approx z)\Big]\Big)$ reduces to $z \not\approx x$, and hence, we have that the above formulas is equivalent to

$$\forall x \forall y \forall z\left[Rxy \wedge Ryz \to \Big[(y \approx x) \vee (y \approx z)\Big] \vee z \not\approx x\right].$$

Rearranging this, we have

$$\forall x \forall y \forall z\left[Rxy \wedge Ryz \wedge z \approx x \to \Big[(y \approx x) \vee (y \approx z)\Big]\right],$$

which reduces to $\forall x \forall y(Rxy \wedge Ryx \to x \approx y)$. This is precisely the definition of anti-symmetry.

Example A.0.3. The formula $p \vee Dp \to \Diamond p$ *(taken from [8])* corresponds to $\forall x \forall y Rxy$.

We first distribute the main implication over the disjunction in the antecedent to get $(p \to \Diamond p) \wedge (Dp \to \Diamond p)$ and deal with each conjunct in turn.

The standard translation of the first conjunct, with free variables universally quantified, is

$$\forall x \forall \mathbf{P} \left[\underbrace{\mathbf{P}x}_{UNI} \to \underbrace{\exists y (Rxy \wedge \mathbf{P}y)}_{POS} \right] .$$

We may apply the minimal instantiation $\sigma(\mathbf{P}) = \lambda u.u \approx x$ to give us

$$\forall x \exists y (Rxy \wedge y \approx x) \ ,$$

which reduces, quite nicely, to $\forall x Rxx$.

The standard translation of the second conjunct, with free variables universally quantified, is given as

$$\forall x \forall \mathbf{P} \left[\exists y (x \not\approx y \wedge \mathbf{P}y) \to \exists z (Rxy \wedge \mathbf{P}y) \right] .$$

We pull out the existential quantifier from the antecedent to get the form

$$\forall x \forall y \forall \mathbf{P} \left[\underbrace{x \not\approx y}_{REL} \wedge \underbrace{\mathbf{P}y}_{UNI} \to \underbrace{\exists z (Rxy \wedge \mathbf{P}y)}_{POS} \right] .$$

Using the minimal instantiation $\sigma(\mathbf{P}) = \lambda u.u \approx y$, we get

$$\forall x \forall y \left[x \not\approx y \to \exists z (Rxz \wedge z \approx y) \right] .$$

This, together with the first conjunct, is equivalent to the first-order property, $\forall x \forall y Rxy$.

Example A.0.4. The formula $\exists p\,(\Box\neg p \wedge \Box\Box p)$ corresponds to

$$\forall x \forall y \forall z (Rxy \wedge Rxz \to \neg Ryz) \ .$$

The standard translation of the globally satisfiability of $\exists p\,(\Box\neg p \wedge \Box\Box p)$ is given by

$$\forall x \exists \mathbf{P} \left[\underbrace{\forall y (Rxy \to \neg \mathbf{P} y)}_{NEG} \wedge \underbrace{\forall z (R^2 xz \to \mathbf{P} z)}_{BOX} \right] \ ,$$

where R^2xz is shorthand for $\exists w(Rxw \wedge Rwz)$.

We read off the minimal instances of $\mathbf{P}$ as $\sigma(\mathbf{P}) = \lambda u.R^2xu$.

Applying Ackermann's Lemma, we have the first-order formula,

$$\forall x \left[\forall y (Rxy \to \neg R^2 xy) \wedge \forall z (R^2 xz \to R^2 xz) \right] \ ,$$

which reduces to $\forall x \forall y \Big(Rxy \to \neg \exists w (Rxw \wedge Rwy) \Big)$.

This further reduces to $\forall x \forall y \forall w (Rxy \wedge Rxw \to Rwy)$, as required.