Algorithmic Correspondence and Completeness in Modal Logic

by

Willem Ernst Conradie

School of Mathematics
University of the Witwatersrand
Johannesburg
South Africa

Under the supervision of
Prof. V. F. Goranko

A thesis submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfillment of the requirements for the degree Doctor of Philosophy.

Johannesburg, 2007

Declaration

I declare that this thesis is my own, unaided work. It is being submitted for the degree of Doctor of Philosophy in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

____

Willem Ernst Conradie

This __________ day of ____________________, at Johannesburg, South Africa.

Acknowledgements

In the first place I want to thank my supervisor Valentin Goranko, with whom it was truly a privilege to work. I have learnt inestimably much from him over the past four years — everything from writing canonicity proofs, to organizing workshops, to making Bulgarian style yogurt.

For their stimulating conversation and ideas I am grateful to André Croucamp, Renate Schmidt, Balder ten Cate, Clint van Alten, Govert van Drimmelen and Dimiter Vakarelov.

I wish to thank Heinrich Raubenheimer and Betsie Jonck for being most understanding and accommodating (in their capacity as heads of department) when it came to negotiating the distribution of my teaching load.

Some of the initial work on the SQEMA-algorithm was done while Valentin Goranko and I were visiting the Department of Computer Science at the University of Manchester. This visit was organized by Renate Schmidt and funded by the UK EPSRC.

I further gratefully acknowledge the financial support of the National Research Foundation of South Africa and of the University of Johannesburg.

Thanks are due to my examiners — Valentin Goranko, Renate Schmidt and Michael Zakharyaschev — for their meticulous reading of the thesis and for their insightful comments and constructive criticism.

Lastly, I would like to thank my family and friends for their continued interest, encouragement, support and patience.

The results reported in this thesis were obtained while I was employed by the University of Johannesburg (former Rand Afrikaans University).

Abstract

This thesis takes an algorithmic perspective on the correspondence between modal and hybrid logics on the one hand, and first-order logic on the other. The canonicity of formulae, and by implication the completeness of logics, is simultaneously treated.

Modal formulae define second-order conditions on frames which, in some cases, are equivalently reducible to first-order conditions. Modal formulae for which the latter is possible are called elementary. As is well known, it is algorithmically undecidable whether a given modal formula defines a first-order frame condition or not. Hence, any attempt at delineating the class of elementary modal formulae by means of a decidable criterium can only constitute an approximation of this class. Syntactically specified such approximations include the classes of Sahlqvist and inductive formulae. The approximations we consider take the form of algorithms.

We develop an algorithm called SQEMA, which computes first-order frame equivalents for modal formulae, by first transforming them into pure formulae in a reversive hybrid language. It is shown that this algorithm subsumes the classes of Sahlqvist and inductive formulae, and that all formulae on which it succeeds are d-persistent (canonical), and hence axiomatize complete normal modal logics.

SQEMA is extended to polyadic languages, and it is shown that this extension succeeds on all polyadic inductive formulae. The canonicity result is also transferred.

SQEMA is next extended to hybrid languages. Persistence results with respect to discrete general frames are obtained for certain of these extensions. The notion of persistence with respect to strongly descriptive general frames is investigated, and some syntactic sufficient conditions for such persistence are obtained. SQEMA is adapted to guarantee the persistence with respect to strongly descriptive frames of the hybrid formulae on which it succeeds, and hence the completeness of the hybrid logics axiomatized with these formulae. New syntactic classes of elementary and canonical hybrid formulae are obtained.

Semantic extensions of SQEMA are obtained by replacing the syntactic criterium of negative/positive polarity, used to determine the applicability of a certain transformation rule, by its semantic correlate — monotonicity. In order to guarantee the canonicity of the formulae on which the thus extended algorithm succeeds, syntactically correct equivalents for monotone formulae are needed. Different version of Lyndon's monotonicity theorem, which guarantee the existence of these equivalents, are proved. Constructive versions of these theorems are also obtained by means of techniques based on bisimulation quantifiers.

Via the standard second-order translation, the modal elementarity problem can be attacked with any second-order quantifier elimination algorithm. Our treatment of this approach takes the form of a study of the DLS-algorithm. We partially characterize the formulae on which DLS succeeds in terms of syntactic criteria. It is shown that DLS succeeds in reducing all Sahlqvist and inductive formulae, and that all modal formulae in a single propositional variable on which it succeeds are canonical.

Contents

Introduction

Correspondence and completeness theory are classical and well-developed areas of modal logic. In this introduction we will briefly sketch a perspective on certain questions from these areas and thus attempt to indicate where the contribution of this thesis lies.

The correspondence between modal languages and predicate logic depends on where one focusses in the multi-layered hierarchy of Kripkean semantics notions. At the bottom of this hierarchy lies the Kripke model. At this level the question of correspondence, at least when approached from the modal side, is trivial: all modal formulae define first-order conditions on these structures.

At the top of the hierarchy, the interpretation of modal languages over Kripke frames turns them into fragments of monadic second-order logic, and rather expressive fragments at that. Indeed, as Thomason ([Tho75]) has shown, second-order consequence may be effectively reduced to the modal consequence over Kripke frames. As is well known, second-order logic is not even recursively axiomatizable.

An intermediate level is offered by general frames. When interpreted on these structures, modal formulae become equivalent to formulae in a two-sorted first-order langauge.

If the question of correspondence being asked, is a question of correspondence with first-order logic, then the interesting level in the hierarchy we have outlined is clearly that of Kripke frames. This is where our efforts are needed in order to try and rescue as much of modal logic as we can from the disadvantages of second-order logic. And indeed, there is much that can be salvaged, for many modal formulae define simple first-order properties of frames. For example, as is well known, the formula $\Box p \rightarrow \Box\Box p$ is valid on precisely the transitive Kripke frames.

By identifying a first-order frame equivalent for a modal formula, we buy for that formula all the advantages that first-order logic has over second-order logic. Specifically, first-order logic is finitely axiomatizable, semi-decidable, compact and admits the Skolem-Löwenheim theorems. In general first-order logic is much better studied than second-order logic and many automated proof tools exist for it.

Hence the question becomes 'which modal formulae define first-order properties of frames?', or more succinctly, 'which modal formulae are elementary?'. There are different forms that an answer to this question can take. From a model theoretic standpoint, a characterization of these formulae might take the form of a list of operations on semantic structures under which their truth is invariant. Elegant such characterizations have been provided by van Benthem ([vB83, vB84]). From a syntactic point of view, descriptions of the shape of formulae that guarantee their first-order definability may be given. As an instance of this type of answer, the class of Sahlqvist formulae ([Sah75]) is probably best known.

But what type of answer will we be satisfied with? Since the computational difficulties associated with second-order logic could be one of our motivations for asking the question of first-order correspondence in the first place, we might like to specify that the answer should offer an effective, i.e. algorithmically verifiable, criterion. Is this a reasonable demand? If we are willing to settle for sufficient conditions this is indeed quite reasonable, but as far as characterizations are concerned, all hopes are dashed by what is known as Chagrova's theorem ([Cha91, CC06]), which states that it is algorithmically undecidable whether a given modal formula defines a first-order condition on Kripke frames.

Hence, if we make decidability a prerequisite, then we will have to be content with approximations of the class of elementary modal formulae. The Sahlqvist formulae and other syntactically specified classes mentioned above are examples of such approximations. Now, the standard proof of the elementarity of the Sahlqvist formulae takes the form of an algorithm, known as the Sahlqvist-van Benthem algorithm, which computes first-order correspondents for the members of this class. However, the syntactic definition of the Sahlqvist formulae is taken as primary. The content of this thesis is an attempt to answer the question: 'What happens if we take the algorithm as primary?' As long as decidability is our goal, this would seem to be the reasonable approach. Approximations of the class of elementary modal formulae would then take the form of classes of formulae for which an algorithm returns the answer 'yes, it is elementary!'. Similarly, semi-algorithms (i.e. procedures not guaranteed to terminate) would define semi-decidable approximations and recursively enumerable sets of elementary modal formulae.

Apart from being elementary, the Sahlqvist formulae have the added virtue of being *canonical* (i.e., of being valid in the canonical, or Henkin, models of the logics axiomatized by them), and hence, of axiomatizing complete normal modal logics. In other words, putting the two properties together, logics axiomatized using Sahlqvist formulae are sound and complete with respect to elementary frame classes. Various results, most notably Fine's theorem ([Fin75b]), the converse of which was recently disproved in [GHV03], link the elementarity and canonicity of modal formulae. It therefore makes sense, when approximating the elementary modal formulae, to simultaneously consider the canonicity of the formulae in the approximating classes. Indeed, canonicity itself is an undecidable property of modal formulae (see e.g. [CZ95] or [Kra99]), and one may therefore wish to make separate decidable approximations of the class of formulae with this property. In so doing one would obtain decidable classes of formulae, the members of which are guaranteed to axiomatize complete logics. Or phrased another way, one would obtain effectively decidable sufficient conditions for the completeness of modal logics. Since completeness is one of the most important question that can be asked about any logic, this might indeed be very useful.

Over the last decade and a half, hybrid logics have become increasingly popular. Hybrid languages enrich traditional modal languages with nominals — a type of propositional variable which acts as a name for a state in a Kripke model — as well as various mechanisms to exploit the naming power of the nominals. The questions of elementarity and canonicity can also be asked of hybrid formulae, and since the undecidability of these properties transfer from the modal case, the program of decidable approximations extends to hybrid languages in a natural way.

Certain fragments of hybrid languages admit of very natural correspondence and canonicity results. These are the fragments consisting of those formulae which contain no propositional variables but, perhaps, nominals — the so-called *pure formulae.* The second-order quantification involved in the interpretation of these formulae on frames, only involves quantification over singleton subsets, and is hence essentially first-order.

An indirect approach to modal (and hybrid) first-order correspondence results thus suggests itself — first try to find a pure equivalent in a, possibly extended, hybrid language, and then rely on the first-order definability of the latter formula. This strategy will be extensively exploited in the ensuing chapters. Incidentally, an application of this approach in the opposite direction, i.e. from first-order to modal formulae via pure hybrid formulae, can be found in [Hod07], where a method for obtaining a (infinite) canonical axiomatization for every modal logic of an elementary frame class is given.

Organization and origins of the content of the thesis

Parts of this thesis are based on (co-authored) papers which have already appeared or have been accepted for publication. Specifically, sections 2.1. up to and including section 2.6. are based upon [CGV06a], a paper co-authored with V. Goranko and D. Vakarelov. Chapter 3 is based on [Con06] (written by the author alone) while Chapter 4 in its entirety is again based upon a paper co-authored with V. Goranko and D. Vakarelov, namely [CGV06b]. The results in chapters 5 and 6 have not appeared elsewhere.

In chapter 0 some essential background on modal and hybrid logic is provided.

Chapter 1 introduces and discusses the notions of the first-order definability and canonicity of modal formulae. Some key results from modal correspondence theory are recalled, and the well known syntactically specified classes of elementary and/or canonical modal formulae are discussed and compared.

Chapter 2 introduces an algorithm, called SQEMA, which computes first-order equivalents for modal formulae by transforming input formulae equivalently into pure hybrid formulae, thus implementing the 'indirect' strategy mentioned above. It is proved that the algorithm is sound, and it is illustrated with a number of examples. It is further shown that all formulae on which the algorithm succeeds have the property which we will take as our definition of canonicity in the modal case, namely d-persistence. The correspondence between modal and pure hybrid formulae which SQEMA exploits is scrutinized more closely. Lastly the algorithm is placed in a broader context by showing that the formulae which are reducible by it are included in the class of van Benthem formulae — a very general, recursively enumerable, but undecidable, class of elementary but not necessarily canonical modal formulae.

Chapter 3 investigates the second-order quantifier elimination algorithm DLS, due to Doherty, Łukaszewicz and Szalas ([DŁS97]). Via the standard second-order translation, the elementarity problem for modal formulae reduces to a second-order quantifier elimination problem. We obtain some partial characterizations of the second-order (and hence modal) formulae on which DLS succeeds. It is shown that all modal formulae in a single propositional variable on (the translation of) which DLS succeeds are canonical.

Chapter 4 extends the SQEMA-algorithm to reversive and polyadic languages. It is shown that the thus extended algorithm successfully computes first-order equivalents for all polyadic inductive formulae.

Chapter 5 extends SQEMA to some hybrid languages. The extended algorithms are designed to guarantee certain persistence (or canonicity) properties of the formulae on which they succeed, which in turn guarantee the completeness of the respective hybrid logics axiomatized with these. Some new syntactically specified classes of elementary and canonical hybrid formulae are introduced.

Chapter 6 explores so-called semantic extensions of SQEMA. Certain operations employed by SQEMA exploit the monotonicity of formulae in given propositional variables. However, the algorithm recognizes the monotonicity (a semantic property) of these formulae, by looking for the stronger, syntactic properties of positivity and negativity. Here we wish to extend the applicability of the algorithm by enabling it to look for monotonicity proper. Methods are developed for computing equivalents, with certain desirable syntactic characteristics, for monotone formulae.

Analysis of results obtained

Elementarity and canonicity are important and interesting properties of modal and hybrid formulae, for reasons outline above. Since both these properties are algorithmically undecidable, one might be interested in decidable approximations of the classes of elementary and/or canonical formulae. The known decidable approximations of the intersection of these classes take the form of syntactically specified classes of formulae, e.g. Sahlqvist formulae, while, in the case of the elementary formulae, second-order quantifier elimination algorithms also provide such approximations.

This thesis develops approximations of (the intersection of) these classes in terms of semi-algorithms, specifically in terms of an algorithm called SQEMA and its extensions and adaptations. The main results and contributions of this thesis can be summed up in terms of the following points:

1. Very general elementarity and completeness results for modal and hybrid formulae, which are shown to subsume many know such results, e.g. those of [Sah75], [GV01], [GV02] and [tCMV05]. Moreover, the classes of formulae for which these results hold are characterized, not in terms of complicated syntactic definitions, but as those formulae that are reducible to pure formulae using a few simple rewrite rules.

2. Algorithmic methods for proving canonicity / completeness of logics.

3. Extensions of these results to reversive and non-reversive polyadic and hybrid languages. Hybrid languages considered include those with and without the universal modality and the satisfaction operator.

4. A partial characterization of the modal formulae and second-order formulae for which the second-order quantifier elimination algorithm DLS ([DŁS97]) succeeds, as well as a limited canonicity result for these formulae. As part of this analysis rigourous proofs that DLS succeeds on all Sahlqvist ([Sah75]) and inductive ([GV01]) formulae are given.

5. Contributions to the completeness theory for hybrid logics:

   (a) Algorithmic and syntactic classes of formulae which axiomatize complete hybrid logics are obtained. These classes subsume different previous such classes (e.g. [GPT87], [GG93], [BT99] and [tCMV05]) and, unlike these classes, allow a more liberal combination of propositional variables and nominals within formulae.

   (b) A study of the notion of sd-persistence (introduced by Ten Cate in [tC05b]) and the development of new classes of sd-persistence formulae.

6. More robust extensions of the SQEMA algorithm which are less dependent on the syntactic shape of input formulae, by using decidable semantic conditions. Results obtained relating to these extension include:

   (a) New versions of Lyndon's theorem ([Lyn59b]) for syntactically open/closed formulae.

   (b) Constructive versions of the above theorems using adaptations of bisimulation quantifiers.

   (c) Extension of the method of (bi)simulation quantifiers (see e.g. [Pit92], [DL02], [Vis96] and [Ghi95]) for constructively obtaining syntactically correct equivalents in reversive languages.

In the light of the above, it may be concluded that algorithmic approaches to questions correspondence and completeness in modal and hybrid logic can indeed be feasible and fruitful.

Chapter 0

Preliminaries

In this chapter we review some essential background knowledge. We focus on the modal and hybrid logics that will be the main players in what is to follow, and also on their relation to first and second-order logic. The interrelationships between these logics will constitute one of the main themes of this thesis. This chapter makes no original contribution, nor does it contain anything unusual or surprising, it merely aims to collect some pertinent facts and to fix some terminology and notation. The reader familiar with these logics might best skip over this chapter entirely, only referring back to it should (s)he ever, in later sections, find him(her)self in doubt as to the meaning of some notation or the precise way in which some term is used.

0.1 Modal logic

Propositional modal languages are obtained by enriching propositional logic with modal operators — in the most basic case, with the dual pair of unary operators $\Diamond$ and $\Box$. These operators can be interpreted in various ways. Modal logic has its origins in philosophy, where modal languages are used to formalize statements involving possibility and necessity ($\Diamond p =$ 'possibly p' and $\Box p =$ 'necessarily p'). Interpreting $\Diamond$ and $\Box$ as 'permissible' and 'obligatory' we obtain the language of deontic logic. The language of tense (or temporal) logic enriches propositional logic with F, P and their duals G and H, and interprets Fp as 'sometime in the future p', Gp as 'always in the future p', Pp as 'sometime in the past p' and Hp as 'always in the past p'. Yet another interpretation uses modal languages to talk about knowledge and belief (epistemic logic). Description logics use modal knowledge representation languages for describing the terminological knowledge of varying application domains. The list goes on.

For this thesis, however, the appropriate perspective is to follow [BdRV01] and regard modal languages, rather generically, as 'simple yet expressive languages for talking about relational structures.'

But so much for the vague and general. In what follows we fix some basic notions of modal logic. For thoroughgoing and up to date treatments of the subject the reader is referred to any one of the standard references [BdRV01], [CZ97] or [Kra99].

0.1.1 Syntax

A *modal similarity type* τ is a pair (O, ρ), where O is a non-empty set of *basic modal terms* and $\rho : O \to \omega$ is an *arity function* assigning to each modal term $\alpha \in O$ a natural number $\rho(\alpha)$, which is the arity of α.

Given a modal similarity type $\tau = (O, \rho)$ and a set of propositional variables Φ, the *modal language* $\mathcal{L}_\tau(\Phi)$ is given by the following recursion:

$$\varphi ::= \bot \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle\alpha\rangle(\varphi_1, \ldots, \varphi_n) \mid [\alpha](\varphi_1, \ldots, \varphi_n)$$

where $p \in \Phi$ and $\alpha \in O$ with $\rho(\alpha) = n$. The boolean connectives $\to$ and $\leftrightarrow$ are defined in terms of $\neg$ and $\vee$ in the usual way, while $\top$ is defined as $\neg\bot$. $[\alpha]$ and $\langle\alpha\rangle$ are called *modalities* or *modal operators*. Specifically, for every $\alpha \in O$, $[\alpha]$ is called a *box modality* (or simply a *box*) and $\langle\alpha\rangle$ is called a *diamond modality* (or simply a *diamond*). $\langle\alpha\rangle$ is the *dual* $[\alpha]$, and vice versa. Note that we take more connectives a primitive than strictly necessary — this will be of technical convenience subsequently.

We will often omit reference to a particular set of propositional variables Φ, and simply write $\mathcal{L}_\tau$, and assume that the language is constructed using some denumerably infinite set of propositional variables. We will write $\mathsf{PROP}(\varphi)$ for the set of all propositional variables that occur in the formula φ. The notation PROP will also sometimes be used to refer to the set of propositional variables over which a language is built.

Modal similarity types τ containing modal terms of arity 1 only called *monadic*, as is any language in such a similarity type. The symbol κ will usually be used to denote an arbitrary monadic similarity type. When dealing with monadic languages, we will often write $\Diamond_\alpha$ and $\Box_\alpha$ in stead of $\langle\alpha\rangle$ and $[\alpha]$, respectively.

Similarity types and languages with terms of higher arity are called *polyadic*. Throughout this thesis we will mostly be working with monadic languages. Polyadic languages have some peculiarities of their own, and are treated (with one or two small exceptions) only in chapter 4. When treating polyadic languages, we will close the set of basic modal terms under composition. This explains the terminology 'basic modal terms', which otherwise might better have been called 'modal indices'.

The *basic modal similarity type* contains only one, unary modal term. The box and diamond modalities corresponding to this term will usually be denoted by $\Box$ and $\Diamond$, respectively. The modal langauge in the basic modal similarity type is called the *basic modal language* and will be denoted simply by $\mathcal{L}$, or by $\mathcal{L}(\Phi)$ when a particular set of propositional variables Φ over which the langauge is built is of importance.

A monadic similarity type κ is called *reversive*, if for every modal term $\alpha \in \kappa$, it is also the case that κ contains the inverse, α^{-1}, of α. In closing a monadic similarity type κ under inverses, we obtain its reversive extension, denoted $r(\kappa)$. We will use the notation $\Diamond_\alpha^{-1}$ interchangeably with $\Diamond_{\alpha^{-1}}$, and similarly with $\Box_\alpha^{-1}$ and $\Box_{\alpha^{-1}}$. In the case of the basic modal langauge $\mathcal{L}$, we will write $\Diamond^{-1}$ and $\Box^{-1}$ for the inverses of $\Diamond$ and $\Box$, respectively. A language over a reversive similarity type is also referred to as a *reversive language*. The *basic reversive langauge* will be denoted by $\mathcal{L}_r$.

For the remainder of this section we will only treat the basic modal similarity type and language. All notions that we will introduce generalize in a straightforward way to arbitrary monadic similarity types. As already remarked, polyadic languages are treated in chapter 4.

0.1.2 Semantics

Modal languages can be interpreted over various, related structures. We will be concerned with frames, models, general frames, and pointed versions of these.

Kripke frames and models

A *(Kripke) frame* is a structure $\mathfrak{F} = (W, R)$, consisting of a non-empty set W of *possible worlds* and, a binary *accessibility relation* between possible worlds $R \subseteq W^2$. Possible worlds are also called *states* or *points* of the frame.

A *Kripke model based on a frame* $\mathfrak{F} = (W, R)$ is a pair $\mathcal{M} = (\mathfrak{F}, V)$ (equivalently, a triple (W, R, V)) where $V : \mathsf{PROP} \longrightarrow 2^W$ is a *valuation* which assigns to every propositional variable the set of possible worlds where it is true. Given two valuations V and V' over the same Kripke frame, we will write $V \sim_p V'$ if V' and V' are identical except, possibly, in the assignment they make to propositional variable p. Valuations V and V' such that $V \sim_p V'$ will be called *p-variants* of each other.

A *pointed frame* $(\mathfrak{F}, w)$ is a pair consisting a frame $\mathfrak{F}$ together with a distinguished point $w \in W$. A pointed model $(\mathcal{M}, w)$ is defined similarly.

The *truth of a formula* $\varphi \in \mathcal{L}$ in a pointed model $(\mathcal{M}, w) = ((W, R, V), w)$, denoted $(\mathcal{M}, w) \Vdash \varphi$, is defined recursively as follows:

- $(\mathcal{M}, w) \nVdash \bot$;
- $(\mathcal{M}, w) \Vdash p$ iff $w \in V(p)$;
- $(\mathcal{M}, w) \Vdash \neg\varphi$ iff $(\mathcal{M}, w) \nVdash \varphi$;
- $(\mathcal{M}, w) \Vdash \varphi \vee \psi$ iff $(\mathcal{M}, w) \Vdash \varphi$ or $(\mathcal{M}, w) \Vdash \psi$;
- $(\mathcal{M}, w) \Vdash \varphi \wedge \psi$ iff $(\mathcal{M}, w) \Vdash \varphi$ and $(\mathcal{M}, w) \Vdash \psi$;
- $(\mathcal{M}, w) \Vdash \Diamond\varphi$ iff there exists a world $u \in W$ such that Rwu and $(\mathcal{M}, u) \Vdash \varphi$;
- $(\mathcal{M}, w) \Vdash \Box\varphi$ iff, for *all* $u \in W$ such that Rwu, it is the case that $(\mathcal{M}, u) \Vdash \varphi$.

In the case of the basic reversive language $\mathcal{L}_r$, the semantics is extended with the clauses:

- $(\mathcal{M}, w) \Vdash \Diamond^{-1}\psi$ iff there exists a world $u \in W$ such that Ruw and $(\mathcal{M}, u) \Vdash \psi$, and
- $(\mathcal{M}, w) \Vdash \Box^{-1}\varphi$ iff, for *all* $u \in W$ such that Ruw, it is the case that $(\mathcal{M}, u) \Vdash \varphi$.

In other words, $\Diamond^{-1}$ and $\Box^{-1}$ are interpreted using the inverse of the accessibility relation R.

A formula φ in is *valid in a model* $\mathcal{M}$, denoted $\mathcal{M} \Vdash \varphi$, if $(\mathcal{M}, w) \Vdash \varphi$ for every $w \in \mathcal{M}$; *valid in a pointed frame* $(\mathfrak{F}, w)$, denoted $(\mathfrak{F}, w) \Vdash \varphi$, if $(\mathcal{M}, w) \Vdash \varphi$ for every model $\mathcal{M}$ based on $\mathfrak{F}$; *valid on a frame* $\mathfrak{F}$, denoted $\mathfrak{F} \Vdash \varphi$, if it is valid in every model based on $\mathfrak{F}$; *valid*, denoted $\Vdash \varphi$, if it is valid on every frame; *globally satisfiable on a frame* $\mathfrak{F}$, if there exists a valuation V such that $(\mathfrak{F}, V) \Vdash \varphi$.

The following notation will often be useful. For a $\varphi \in \mathcal{L}_r$ and a model $\mathcal{M}$ we write $[\![\varphi]\!]_{\mathcal{M}} = \{w \in \mathcal{M} \mid (\mathcal{M}, w) \Vdash \varphi\}$ for the *extension* (or *truth-set*) of φ in $\mathcal{M}$.

General frames

A *general frame* is a structure $\mathfrak{g} = (W, R, \mathbb{W})$ where (W, R) is a frame, and $\mathbb{W}$ is a Boolean algebra of subsets of 2^W, called *the admissible sets* in $\mathfrak{g}$, also closed under the modal operator $\Diamond$, defined as follows:

$$\Diamond X = \{y \in W \mid Ryx \text{ for some } x \in X\}.$$

Clearly, $\mathbb{W}$ is also closed under the dual operators $\Box$, defined accordingly:

$$\Box X = \{y \in W \mid Ryx \text{ implies } x \in X\}.$$

The alternative notation m_R and l_R for $\Diamond$ and $\Box$, respectively, is sometimes preferable, if we wish to distinguish between the modal syntax and the operators on algebras more explicitly.

General reversive frames are defined analogously, adding the additional requirement that the algebra of admissible sets also be closed under the $\Diamond^{-1}$ and $\Box^{-1}$-operators.

Extending the correlation between the modal box and diamond and operators on the algebras of admissible sets of general frames, we can regard any modal formula as such an operator. Specifically, given a general frame $\mathfrak{g} = (\mathfrak{F}, \mathbb{W})$, admissible sets $A_1, \ldots, A_n \in \mathbb{W}$ and a formula φ with $\mathsf{PROP}(\varphi) = \{p_1, \ldots, p_n\}$, we write $\varphi(A_1, \ldots, A_n)$ for the set $[\![\varphi]\!]_{(\mathfrak{F},V)}$ where V is any valuation assigning A_i to p_i, $1 \leq i \leq n$.

The *underlying Kripke frame* of a general frame $\mathfrak{g} = (W, R, \mathbb{W})$ is the frame (W, R), denoted $\mathfrak{g}_\sharp$, i.e. the underlying Kripke frame is obtained by forgetting about the algebra of admissible sets. A *model over* $\mathfrak{g}$ is a model over $\mathfrak{g}_\sharp$ with the valuation of the variables ranging over $\mathbb{W}$. All notions of local and global truth, validity and satisfiability of formulae are accordingly relativized with respect to general frames and models based on them.

Here are a few types of general frames that will be encountered further on. A general frame $\mathfrak{g} = (W, R, \mathbb{W})$ is said to be:

differentiated if for every $x, y \in W$, $x \neq y$, there exists $X \in \mathbb{W}$ such that $x \in X$ and $y \notin X$;

tight if for all $x, y \in W$ it is the case that Rxy iff $x \in \bigcap\{\langle R\rangle(Y) \mid Y \in \mathbb{W} \text{ and } y \in Y\}$;

compact if every family of admissible sets from $\mathbb{W}$ with the finite intersection property (FIP) has non empty intersection (recall that a family of sets has the finite intersection property if every finite subfamily has non-empty intersection);

refined if it is differentiated and tight;

descriptive it is refined and compact;

discrete if every singleton subset of W is in $\mathbb{W}$.

Definition 0.1.1 Given a class of general frames $\mathfrak{C}$, a formula φ is *locally* $\mathfrak{C}$*-persistent* if $(\mathfrak{g}, w) \Vdash \varphi$ implies $(\mathfrak{g}_\sharp, w) \Vdash \varphi$ for every pointed general frame $(\mathfrak{g}, w)$ in $\mathfrak{C}$. A formula φ is *(globally)* $\mathfrak{C}$*-persistent* if $\mathfrak{g} \Vdash \varphi$ implies $\mathfrak{g}_\sharp \Vdash \varphi$ for every general frame $\mathfrak{g}$ in $\mathfrak{C}$. Clearly local $\mathfrak{C}$-persistence implies global $\mathfrak{C}$-persistence.

Specifically, a formula is (locally) *r-persistent* of it is (locally) persistent with respect to the class of all refined general frames, (locally) *d-persistent* of it is (locally) persistent with respect to the class of all descriptive general frames, and (locally) *di-persistent* of it is (locally) persistent with respect to the class of discrete general frames.

Consequence relations and equivalence notions

The semantic structures defined above give rise to various notions of consequence. Let $\mathfrak{C}$ be a class of models, frames or general frames. We say a formula φ is a *local consequence* of a set of formulae Γ over $\mathfrak{C}$ if $(M, m) \Vdash \Gamma$ implies $(M, m) \Vdash \varphi$, for every member M of $\mathfrak{C}$ and every point $m \in M$. We write $\Gamma \Vdash^{loc}_{\mathfrak{C}} \varphi$. When $\mathfrak{C}$ is the class of all models, general frames or frames we write $\Gamma \Vdash^{loc}_{\mathrm{mod}} \varphi$, $\Gamma \Vdash^{loc}_{\mathrm{gf}} \varphi$ and $\Gamma \Vdash^{loc}_{\mathrm{fr}} \varphi$, respectively. We will usually simply write $\Gamma \Vdash \varphi$ for $\Gamma \Vdash^{loc}_{\mathrm{mod}} \varphi$. Similarly φ is a *global consequence* of Γ over $\mathfrak{C}$ if $M \Vdash \Gamma$ implies $M \Vdash \varphi$, for every member M of $\mathfrak{C}$. Local consequence clearly implies global consequence. Again we write $\Gamma \Vdash^{gl}_{\mathrm{mod}} \varphi$, $\Gamma \Vdash^{gl}_{\mathrm{gf}} \varphi$ and $\Gamma \Vdash^{gl}_{\mathrm{fr}} \varphi$ in the cases when $\mathfrak{C}$ is the class of all models, general frames and frames, respectively.

In all the cases above we will write $\varphi \Vdash \psi$ for $\{\varphi\} \Vdash \psi$ and $\varphi \dashv\Vdash \psi$ when $\varphi \Vdash \psi$ and $\psi \Vdash \varphi$. Accompanying these consequence relations are a number of equivalence notions, for which we introduce the following, more convenient terminology. Formulae φ and ψ are:

semantically equivalent, denoted $\varphi \equiv_{\mathrm{sem}} \psi$, if φ and ψ are true at exactly the same points in the same models, i.e. when $\varphi \dashv\Vdash^{loc}_{\mathrm{mod}} \psi$;

model-equivalent, denoted $\varphi \equiv_{\mathrm{mod}} \psi$, if φ and ψ are valid on exactly the same models, i.e when $\varphi \dashv\Vdash^{glo}_{\mathrm{mod}} \psi$;

locally equivalent, denoted $\varphi \equiv_{\mathrm{loc}} \psi$, if φ and ψ are valid at exactly the same points in the same general frames, i.e. when $\varphi \dashv\Vdash^{loc}_{\mathrm{gf}} \psi$;

axiomatically equivalent, denoted $\varphi \equiv_{\mathrm{ax}} \psi$, if φ and ψ are valid in the same general frames, i.e when $\varphi \dashv\Vdash^{gl}_{\mathrm{gf}} \psi$. (The reason for this terminology will become apparent later.);

locally frame-equivalent, , denoted $\varphi \equiv_{\mathrm{lfr}} \psi$, if φ and ψ are valid at exactly the same points in the same frames, i.e. when $\varphi \dashv\Vdash^{loc}_{\mathrm{fr}} \psi$;

frame-equivalent, denoted $\varphi \equiv_{\mathrm{fr}} \psi$, if φ and ψ are valid on exactly the same frames, i.e. when $\varphi \dashv\Vdash^{gl}_{\mathrm{fr}} \psi$.

The following relationships hold between these equivalence notions.

$$(\varphi \equiv_{\mathrm{sem}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{mod}} \psi) \ \text{ and } \ (\varphi \equiv_{\mathrm{loc}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{ax}} \psi) \ \text{ and } \ (\varphi \equiv_{\mathrm{lfr}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{fr}} \psi),$$

$$(\varphi \equiv_{\mathrm{mod}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{ax}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{fr}} \psi),$$

and

$$(\varphi \equiv_{\mathrm{sem}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{loc}} \psi) \Longrightarrow (\varphi \equiv_{\mathrm{lfr}} \psi).$$

In other words, for each type of structure, the local equivalence notion implies the accompanying global one. Also, for both the local and global equivalence notions, equivalence on models implies equivalence on general frames, which in turn implies equivalence on frames. Moreover, none of the converses of these implications hold, as illustrated by the following counterexamples:

- The formulae $p \to \Diamond p$ and $(p \to \Diamond p) \wedge \Box(p \to \Diamond p)$ are model, axiomatically and frame-equivalent, yet they are not locally frame-equivalent, and hence neither semantically nor locally equivalent.

- The formulas $p \to \Diamond p$ and $q \to \Diamond q$ are axiomatically equivalent (both axiomatizing the logic **T**), but are clearly not modal-equivalent.

- This example is due to van Benthem [vB84], and can also be found in [BdRV01]. Let $\varphi = \Box\Diamond\top \to \Box(\Box(\Box p \to p) \to p)$ and $\psi = \Box\Diamond\top \to \Box\bot$. It can be shown that φ and ψ are frame equivalent. However, there is a general frame validating φ but on which ψ can be refuted.

- The formulas p and q are obviously not semantically equivalent, yet they are locally equivalent, as neither is valid at any point in any general frame.

- The formulae $\Box(\Box p \leftrightarrow p) \to \Box p$ and $\Box(\Box p \to p) \to \Box p$ are locally frame equivalent but not locally equivalent. This example is from [BS85].

Standard translation

Define L_0 to be the first-order language with $=$, a binary relation symbol R, and individual variables $\mathsf{VAR} = \{x_0, x_1, \ldots\}$. Also, let L_1 be the extension of L_0 with a set of unary predicates $\{P_0, P_1, \ldots\}$ corresponding to the propositional variables $\{p_0, p_1, \ldots\}$. $\mathcal{L}$-formulae are translated into L_1 by means of the following *standard translation* function $\mathrm{ST}(\cdot,\cdot)$ which takes as arguments an $\mathcal{L}$-formula together with a variable from VAR:

- $\mathrm{ST}(\bot, x) := x \neq x$
- $\mathrm{ST}(p_i, x) := P_i(x)$ for every propositional variable p_i;
- $\mathrm{ST}(\neg\varphi, x) := \neg\mathrm{ST}(\varphi, x)$;
- $\mathrm{ST}(\varphi \vee \psi, x) := \mathrm{ST}(\varphi, x) \vee \mathrm{ST}(\psi, x)$;
- $\mathrm{ST}(\varphi \wedge \psi, x) := \mathrm{ST}(\varphi, x) \wedge \mathrm{ST}(\psi, x)$;
- $\mathrm{ST}(\Diamond\varphi, x) := \exists y(Rxy \wedge \mathrm{ST}(\varphi, y))$, where y is the first variable in VAR not appearing in $\mathrm{ST}(\varphi, x)$;
- $\mathrm{ST}(\Box\varphi, x) := \forall y(\neg Rxy \vee \mathrm{ST}(\varphi, y))$, where y is as in the previous item.

We extend $\mathrm{ST}(\cdot,\cdot)$ to $\mathcal{L}_r$ by adding the clauses

$$\mathrm{ST}(\Diamond^{-1}\varphi, x) := \exists y(Ryx \wedge \mathrm{ST}(\varphi, y)),$$

and

$$\mathrm{ST}(\Box^{-1}\varphi, x) := \forall y(\neg Ryx \vee \mathrm{ST}(\varphi, y)),$$

where y is again the first variable in VAR not appearing in $\mathrm{ST}(\varphi, x)$.

Of course, a (modal) model is nothing but an L_1-structure and a Kripke frame nothing but a L_0-structure. In fact we have:

Proposition 0.1.2 *For any pointed model $\mathcal{M}$ and formula $\varphi \in \mathcal{L}$*

$$(\mathcal{M}, m) \Vdash \varphi \quad \textit{iff} \quad \mathcal{M} \models \mathrm{ST}(\varphi, x)[x := w].$$

Hence, for all pointed frames $(\mathfrak{F}, w)$ and formulae $\varphi \in \mathcal{L}$, we have that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathfrak{F} \models \forall\overline{P}\mathrm{ST}(\varphi, x)[x := w]$, and $\mathfrak{F} \Vdash \varphi$ iff $\mathfrak{F} \models \forall x\forall\overline{P}\mathrm{ST}(\varphi, x)$ where $\overline{P}$ is the vector of all predicate symbols corresponding to propositional variables occurring in φ.

0.1.3 Logics

Definition 0.1.3 A *normal modal logic* (in the basic modal languages) is a set of $\mathcal{L}$-formulae Λ, such that

1. Λ contains all propositional tautologies,
2. Λ contains the axiom $\Box(p \to q) \to (\Box p \to \Box q)$ (known as K) and $\Diamond p \leftrightarrow \neg\Box\neg p$ (the dual axiom),
3. Λ is closed under *uniform substitution* (if $\varphi \in \Lambda$ than all substitution instances of φ are in Λ), under *modus ponens* ($\varphi \to \psi, \varphi \in \Lambda$ implies $\psi \in \Lambda$), and under *necessitation* ($\varphi \in \Lambda$ implies $\Box\varphi \in \Lambda$.)

For any set of formulas Γ, there is a smallest normal modal logic containing it. The minimal normal modal logic is called **K**, in honour of Kripke. We will not deal with non-normal logics, so henceforth 'logic' will mean 'normal modal logic'.

When working in $\mathcal{L}_r$ we have to add the additional axioms $\Box^{-1}(p \to q) \to (\Box^{-1}p \to \Box^{-1}q)$ (the K-axiom for the inverse modality), $\Diamond^{-1}p \leftrightarrow \neg\Box^{-1}\neg p$ (the dual-axiom for $\Diamond^{-1}$) as well as the *converse axioms* $p \to \Box\Diamond^{-1}p$ and $p \to \Box^{-1}\Diamond p$, as well as a version of the necessitation rule for $\Box^{-1}$. The minimal logic so obtained, which we will denote by $\mathbf{K}_r$, is called the *minimal tense logic*. The minimal logics corresponding to languages in other, multi-modal and polyadic similarity types are the obvious generalizations of $\mathbf{K}$ and $\mathbf{K}_r$.

Given a logic Λ and a formula $\varphi \in \Lambda$, we write $\vdash_\Lambda \varphi$ and call φ a *theorem* of Λ . For a set of formulae Γ we write $\Gamma \vdash_\Lambda \varphi$ if there are $\varphi_1, \ldots, \varphi_n \in \Gamma$ such that $\vdash_\Lambda (\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi$. A set of formulae Γ is Λ-*consistent* if $\Gamma \not\vdash_\Lambda \bot$ and Λ-*inconsistent* otherwise.

Given a logic Λ and a set of formula Γ we will denote the minimal logic containing $\Lambda \cup \Gamma$ by $\Lambda \oplus \Gamma$.

Given a class $\mathfrak{C}$ of frames or of general frames, the set of all formulae valid in all members of $\mathfrak{C}$ forms a logic, namely the *logic of* $\mathfrak{C}$, which we will denote $\Lambda_{\mathfrak{C}}$. The class of all Kripke frames (respectively, general frames) which validate all formulae in a logic Λ, is called the class of *Kripke frames (respectively, general frames) of* Λ.

Definition 0.1.4 A logic Λ is *sound* with respect to a class of structures $\mathfrak{C}$ is $\Lambda \subseteq \Lambda_{\mathfrak{C}}$. Λ is *complete* with respect to $\mathfrak{C}$ if $\Lambda_{\mathfrak{C}} \subseteq \Lambda$. Λ is *strongly complete* with respect to $\mathfrak{C}$ if $\Gamma \vdash_{\Lambda} \varphi$ whenever $\Gamma \Vdash^{loc}_{mod(\mathfrak{C})} \varphi$, where $mod(\mathfrak{C})$ is the class of models based on structures in $\mathfrak{C}$. We will say a logic is (strongly) complete if it is (strongly) complete with respect to its class of Kripke frames.

As will be shown in chapter 1, we have the following general completeness result with respect to general frames.

Theorem 0.1.5 *Every logic is sound and strongly complete with respect to its class of general frames.*

0.2 Hybrid logic

The possible world, or state, is central to the semantics of modal logic. Yet, the syntax of the languages in the previous section contain no mechanism by which one can refer to states directly. Hybrid logic addresses this situation by adding to the language a special sort of propositional variable, called a nominals, together with the semantic condition that valuations are always to assign singleton subsets of the domain to nominals. A nominal is thus a type of constant, and hence acts as a name for a state in a model.

0.2.1 Syntax and semantics

Let PROP and NOM be disjoint sets, respectively of propositional variables and nominals. We will denote nominals with boldface letters $\mathbf{i}, \mathbf{j}, \mathbf{k}, \ldots$, possibly indexed. The basic hybrid language $\mathcal{L}^n(\mathsf{PROP}, \mathsf{NOM})$ extends the basic modal language $\mathcal{L}(\mathsf{PROP})$ with the clause that every nominal from NOM is a formula. As with $\mathcal{L}$, we usually omit reference to PROP and NOM and write simply $\mathcal{L}^n$ for $\mathcal{L}^n(\mathsf{PROP}, \mathsf{NOM})$. This language is interpreted over the same structures as $\mathcal{L}$, with the additional requirement that valuations now also assign subsets of the domain to nominals, specifically, singleton subsets. Let $\mathcal{M} = (W, R, V)$ be a model. The truth definition is extended with the clause

- $(\mathcal{M}, m) \Vdash \mathbf{i}$ iff $V(\mathbf{i}) = \{m\}$.

All notions of local and global truth, validity and consequence are the obvious generalizations of those for $\mathcal{L}$. The language $\mathcal{L}^n_r$ is obtained by similarly enriching $\mathcal{L}_r$ with nominals.

Hybrid languages are usually further equipped with either the universal modality $[\mathsf{u}]$, or with satisfaction operators $@_{\mathbf{i}}, @_{\mathbf{j}}, \cdots$. The language obtained in the first case is called $\mathcal{L}^{n,u}$ while the second is denoted $\mathcal{L}^{n,@}$. A hybrid formula is *pure* if it contains no propositional variables. Pure formulae are allowed to contain nominals.

The universal modality $[\mathsf{u}]$ and its dual $\langle\mathsf{u}\rangle$ behave syntactically just like an ordinary box-diamond pair, but we specify that the accessibility relation used to interpret it must always be the universal relation. That is to say,

- $(\mathcal{M}, m) \Vdash [\mathsf{u}]\varphi$ iff $(\mathcal{M}, w) \Vdash \varphi$ for all $w \in \mathcal{M}$.

The idea behind the satisfaction operators is that $@_{\mathbf{i}}\varphi$ should express the fact that φ holds at the world named by $\mathbf{i}$. Formally, if $\mathcal{M} = (W, R, V)$ is a model, then

- $(\mathcal{M}, m) \Vdash @_{\mathbf{i}}\varphi$ iff $(\mathcal{M}, w) \Vdash \varphi$ where $V(\mathbf{i}) = \{w\}$.

The language $\mathcal{L}^{n,u}$ is at least as expressive as $\mathcal{L}^{n,@}$, for note that

$$(\mathcal{M}, m) \Vdash @_{\mathbf{i}}\varphi \;\text{ iff }\; (\mathcal{M}, m) \Vdash [\mathsf{u}](\mathbf{i} \to \varphi) \;\text{ iff }\; (\mathcal{M}, m) \Vdash \langle\mathsf{u}\rangle(\mathbf{i} \wedge \varphi).$$

Both the universal modality and the satisfaction operator import a global flavour into our otherwise essentially local semantics. Indeed,

$$(\mathcal{M}, m) \Vdash [\mathsf{u}]\varphi \;\text{ iff }\; \mathcal{M} \Vdash [\mathsf{u}]\varphi$$

and

$$(\mathcal{M}, m) \Vdash @_{\mathbf{i}}\varphi \;\text{ iff }\; \mathcal{M} \Vdash @_{\mathbf{i}}\varphi.$$

Given a frame $\mathfrak{F} = (W, R)$, points $w, w_1, \ldots, w_n \in W$ and subsets $A_1, \ldots A_m \subseteq W$, we say that a formula $\varphi \in \mathcal{L}_r^n$ is $[p_1 := A_1, \ldots, p_m := A_m, \mathbf{i}_1 := w_1, \ldots, \mathbf{i}_n := w_n]$-satisfiable at $(\mathfrak{F}, w)$ if there exists a valuation V on $\mathfrak{F}$ such that $V(p_i) = A_i$, $1 \leq i \leq m$, $V(\mathbf{i}_i) = \{w_i\}$, $1 \leq i \leq n$ and $((\mathfrak{F}, V), w) \Vdash \varphi$. This type of satisfiability will be refereed to as *parameterized satisfiability*. Global parameterized satisfiability is defined analogously. In practice, however, this notation will be much more compact, as we will usually parameterize with only one or at most two propositional variables or nominals.

The standard translation function $\mathrm{ST}(\cdot,\cdot)$ is extended to (the sublanguages of) $\mathcal{L}_r^{n,u}$ and $\mathcal{L}_r^{n,@}$ by the addition of the clauses

- $\mathrm{ST}(\mathbf{i}, x) := y_{\mathbf{i}} = x$ where $y_{\mathbf{i}}$ is a reserved variable associated with the nominal $\mathbf{i}$,

- $\mathrm{ST}(\langle\mathsf{u}\rangle\varphi, x) := \exists y \mathrm{ST}(\varphi, y)$ where y is the first variable not occurring in $\mathrm{ST}(\varphi, x)$,

- $\mathrm{ST}([\mathsf{u}]\varphi, x) := \forall y \mathrm{ST}(\varphi, y)$ where y is in the previous item, and

- $\mathrm{ST}(@_{\mathbf{i}}\varphi, x) := \forall y(y \neq y_{\mathbf{i}} \vee \mathrm{ST}(\varphi, y))$ where y and $y_{\mathbf{i}}$ are as before.

Another, perhaps more natural, option is to translate nominals using individual constants, but because this would take us (at least syntactically) outside L_0, we prefer to use variables.

0.2.2 Logics

We now recall axiomatizations of the minimal logics in the languages $\mathcal{L}^n$, $\mathcal{L}^{n,u}$ and $\mathcal{L}^{@}$. The systems given here are based on those in [BdRV01] and [tC05b]. The original, equivalent axiomatizations of $\mathcal{L}^n$ and $\mathcal{L}^{n,u}$ were first given in [GPT87] and [GG93], respectively. Instead on the *Name* and *Paste*-rules below, these axiomatizations use a rule based on so-called *necessity forms*. We will use the notation $\Diamond^n$ with $n \in \omega$ to denote a string of n consecutive $\Diamond$'s. The notation $\Box^n$ is defined similarly.

Definition 0.2.1 The logic $\mathbf{K}^n$ is the smallest set of $\mathcal{L}^n$-formulae containing all propositional tautologies as well as the common axioms in table 1, and which is closed under the common rules of inference in table 2.

The logic $\mathbf{K}^{n,u}$ is the smallest set of $\mathcal{L}^{n,u}$-formulae containing all propositional tautologies, the common axioms and axioms for $[\mathsf{u}]$ in table 1, and which is closed under the common rules of inference as well as the rules for $[\mathsf{u}]$ in table 2.

The logic $\mathbf{K}^{n,@}$ is the smallest set of $\mathcal{L}^{n,@}$-formulae containing all propositional tautologies, the common axioms and axioms for @ in table 1, and which is closed under the common rules of inference as well as the rules for @ in table 2.

The $Name$ and $Paste$-rules and their $[\mathsf{u}]$ and @ versions are known as called 'non-orthodox' rules, because of their syntactic side-conditions. It is well known that these rules are admissible in the minimal hybrid logics obtained by omitting them. However, they are needed in order to obtain the following well-known general completeness result regarding extensions with pure axioms.

Theorem 0.2.2 ([GPT87, GG93, BT99]) *Let Γ be a set of pure $\mathcal{L}^n$-formulae ($\mathcal{L}^{n,u}$-formulae, respectively $\mathcal{L}^{n,@}$-formulae). Then the logic $\mathbf{K}^n \oplus \Gamma$ ($\mathbf{K}^{n,u} \oplus \Gamma$, respectively $\mathbf{K}^{n,@} \oplus \Gamma$) is strongly sound and complete with respect to its class of Kripke frames.*

For $\mathcal{L}^n$ this result was first-proved in [GPT87]. The case for $\mathcal{L}^{n,u}$ was dealt with in [GG93] and adapted to $\mathcal{L}^{n,@}$ in [BT99].

Common axioms	
(Taut)	All propositional tautologies
(K)	$\Box(p \to q) \to (\Box p \to \Box q)$
(Dual)	$\Diamond p \leftrightarrow \neg\Box\neg p$
(Nom)	$\Diamond^n(\mathbf{i} \wedge p) \to \Box^m(\mathbf{i} \to p)$ for all $0 \leq n, m$
Axioms for $[\mathsf{u}]$	
(K_{U})	$[\mathsf{u}](p \to q) \to ([\mathsf{u}]p \to [\mathsf{u}]q)$
($\mathrm{Dual}_{\mathsf{U}}$)	$\langle\mathsf{u}\rangle p \leftrightarrow \neg[\mathsf{u}]\neg p$
($\mathrm{Ref}_{\mathsf{U}}$)	$[\mathsf{u}]p \to p$
($\mathrm{Sym}_{\mathsf{U}}$)	$p \to [\mathsf{u}]\langle\mathsf{u}\rangle p$
($\mathrm{Trans}_{\mathsf{U}}$)	$[\mathsf{u}]p \to [\mathsf{u}][\mathsf{u}]p$
($\mathrm{Incl}_{\mathsf{U}}$)	$[\mathsf{u}]p \to \Box p$
($\mathrm{Incl}_{\mathbf{i}}$)	$\langle\mathsf{u}\rangle\mathbf{i}$
($\mathrm{Nom}_{\mathbf{i}}$)	$\langle\mathsf{u}\rangle(\mathbf{i} \wedge p) \to [\mathsf{u}](\mathbf{i} \to p)$
Axioms for @	
($K_{@}$)	$@_{\mathbf{i}}(p \to q) \to (@_{\mathbf{i}}p \to @_{\mathbf{i}}q)$
(Self-dual)	$@_{\mathbf{i}}p \leftrightarrow \neg@_{\mathbf{i}}\neg p$
(Introduction)	$\mathbf{i} \wedge p \to @_{\mathbf{i}}p$
(Ref)	$@_{\mathbf{i}}\mathbf{i}$
(Sym)	$@_{\mathbf{i}}\mathbf{j} \leftrightarrow @_{\mathbf{j}}\mathbf{i}$
($\mathrm{Nom}_{@}$)	$@_{\mathbf{i}}\mathbf{j} \wedge @_{\mathbf{j}}p \to @_{\mathbf{i}}p$
(Agree)	$@_{\mathbf{j}}@_{\mathbf{i}}p \leftrightarrow @_{\mathbf{i}}p$
(Back)	$\Diamond @_{\mathbf{i}}p \to @_{\mathbf{i}}p$

Table 1: Axioms for Hybrid Logics

Common rules	
(Modus ponens)	If $\vdash \varphi \to \psi$ and $\vdash \varphi$ then $\vdash \psi$
(Sorted substitution)	$\vdash \varphi'$ whenever and $\vdash \varphi$ and φ' is obtained from φ by uniform substitution of formulas for propositional variables and nominals for nominals.
(Necessitation)	If $\vdash \varphi$ then $\vdash \Box\varphi$.
(Name)	If $\vdash \mathbf{i} \to \varphi$ then $\vdash \varphi$, for $\mathbf{i} \notin \mathsf{NOM}(\varphi)$.
(Paste)	If $\vdash \Diamond^n(\mathbf{i} \wedge \Diamond(\mathbf{j} \wedge \varphi)) \to \psi$ then $\vdash \Diamond^n(\mathbf{i} \wedge \Diamond\varphi) \to \psi$ where $0 \leq n$, $\mathbf{i} \neq \mathbf{j}$ and $\mathbf{j} \notin \mathsf{NOM}(\varphi) \cup \mathsf{NOM}(\psi)$.
Rules for $[\mathsf{u}]$	
($[\mathsf{u}]$-necessitation)	If $\vdash \varphi$ then $\vdash [\mathsf{u}]\varphi$.
(Paste$_{\mathsf{U}}$)	If $\vdash \langle\mathsf{u}\rangle(\mathbf{i} \wedge \Diamond\mathbf{j}) \wedge \langle\mathsf{u}\rangle(\mathbf{j} \wedge \varphi) \to \psi$ then $\vdash \langle\mathsf{u}\rangle(\mathbf{i} \wedge \Diamond\psi) \to \psi$ where $\mathbf{i} \neq \mathbf{j}$ and $\mathbf{j} \notin \mathsf{NOM}(\varphi) \cup \mathsf{NOM}(\psi)$.
Rules for @	
(@-necessitation)	If $\vdash \varphi$ then $\vdash @_{\mathbf{i}}\varphi$ for any nominal $\mathbf{i}$.
(Name$_{@}$)	If $\vdash @_{\mathbf{i}}\varphi$, then $\vdash \varphi$, for $\mathbf{i} \notin \mathsf{NOM}(\varphi)$.
(Paste$_{@}$)	If $\vdash @_{\mathbf{i}}\Diamond\mathbf{j} \wedge @_{\mathbf{j}}\varphi \to \psi$ then $\vdash @_{\mathbf{i}}\Diamond\varphi \to \psi$ where $\mathbf{i} \neq \mathbf{j}$ and $\mathbf{j} \notin \mathsf{NOM}(\varphi) \cup \mathsf{NOM}(\psi)$.

Table 2: Rules of deduction for hybrid logics

Chapter 1

Correspondence and Canonicity

In this chapter we take whirlwind tour through some aspects of modal correspondence and completeness theory. There is a vast literature on these topics — we will not even try to mention all of the highlights. It will be like one of those package tours that 'does Europe' in five days: we will make ten minute stops at the Colosseum, the Eiffel tower and Big Ben. We will only drive past the Louvre and the Vatican in the bus. We will probably skip Switzerland and half a dozen other countries altogether.

In section 1.1 we start with the idea that modal formulae correspond to second and sometimes first-order conditions on frames. We recall that the class of elementary modal formulae is undecidable, but also quote an elegant model-theoretic characterization of this class. In section 1.2 we recall the general method for proving completeness for modal logics which uses the canonical model construction, and also what is meant by a *canonical logic / formula*. In section 1.3 we slacken the pase and take some time to consider the best know syntactically specified classes of elementary and canonical modal formulae in some detail. In section 1.4 we discuss two second-order quantifier elimination algorithms which can be used to obtain first-order equivalents for modal formulae. We conclude in section 1.5 by considering the closure of some syntactically specified classes of elementary and canonical formulae under different notions of equivalence.

1.1 Correspondence with first-order logic

When we introduced the semantics of modal logic in chapter 0, we saw that any modal formula φ defines a first-order condition on models, viz. $\mathcal{M} \Vdash \varphi$ if and only if $\mathcal{M} \models \forall x \mathrm{ST}(\varphi, x)$. However, changing perspective to the frames upon which our models are built, we find that the notion of frame validity involves quantification over valuations, i.e. over subsets of the domain, yielding second-order conditions. That is to say, $\mathfrak{F} \Vdash \varphi$ if and only if $\mathfrak{F} \models \forall P_1 \ldots \forall P_n \forall x \mathrm{ST}(\varphi, x)$, where $P_1, \ldots, P_n$ are the predicate variables corresponding to the propositional variables occurring in φ.

In this way, the formula $\Box p \to p$ is valid on a frame $\mathfrak{F}$ if and only of $\mathfrak{F} \models \forall P \forall x (\forall y (Rxy \to P(y)) \to P(x))$, imposing, what seems to be, a second-order condition on the accessibility relation R of $\mathfrak{F}$. But it turns out that we can show that $\mathfrak{F} \Vdash \Box p \to p$ iff the accessibility relation R of $\mathfrak{F}$ is reflexive — a simple first-order condition! To see why this is so, but also to illustrate some typical arguments used to establish such first-order correspondences, we enshrine this fact as a proposition and give a proof.

Proposition 1.1.1 *A frame $\mathfrak{F}$ validates the formula $\Box p \to p$ if and only if it is reflexive.*

Proof. Suppose that $\mathfrak{F} = (W, R)$ contains a point w such that $\mathfrak{F} \models \neg Rww$. Let S be the set of all successors of w. Let V be a valuation on $\mathfrak{F}$ such that $V(p) = S$. Then $((\mathfrak{F}, V), w) \Vdash \Box p$ but, since $w \notin S$, $((\mathfrak{F}, V), w) \Vdash \neg p$. Hence $\mathfrak{F} \nVdash \Box p \to p$.

Conversely, suppose that R is reflexive. Then for any point $w \in \mathfrak{F}$, any valuation V making $\Box p$ true at w will have to be such that $S \subseteq V(p)$, where S is the set of successors of w. But then $w \in V(p)$ since $w \in S$, and hence $((\mathfrak{F}, V), w) \Vdash p$. QED

Note that the proof actually gives us something stronger, namely that $(\mathfrak{F}, w) \Vdash \Box p \to p$ iff $\mathfrak{F} \models Rxx[x := w]$. In other words, even when taking a local perspective, there is a first-order formula corresponding to $\Box p \to p$. We formalize these ideas in the following definition.

Definition 1.1.2 An L_0-formula $\alpha(x)$ in one free variable is a *local frame correspondent* of a modal formula φ if, for all frames $\mathfrak{F}$ and points $w \in \mathfrak{F}$,

$$(\mathfrak{F}, w) \Vdash \varphi \quad \text{iff} \quad \mathfrak{F} \models \alpha(x)[x := w].$$

Similarly, an L_0-sentence α is a *(global) frame correspondent* of a modal formula φ if

$$\mathfrak{F} \Vdash \varphi \quad \text{iff} \quad \mathfrak{F} \models \alpha.$$

A modal formula φ is *locally first-order definable* or *locally elementary* if has a local first-order frame correspondent. Similarly, formulae with (global) first-order frame correspondents are called *(globally) first-order definable* or *(globally) elementary*.

If $\alpha(x)$ is a local first-order frame correspondent for φ, then $\forall x \alpha(x)$ is a global correspondent for it. So local elementarity implies global elementarity. The converse does not hold: it is easy to see, using reasoning similar to that employed in the proof of proposition 1.1.1, that $\Box\Diamond\Box\Box p \to \Diamond\Diamond\Box\Diamond p$ globally corresponds on frames to $\forall x \exists y Rxy$. However, in [vB83] van Benthem shows that when interpreted locally, this formula violates the Löwenheim-Skolem theorem, and that hence is can have no local first-order frame correspondent.

This result also provides a negative answer to the question as to whether all modal formulae are locally elementary. So are all modal formulae perhaps globally elementary? Once again the answer is 'no' — neither the McKinsey axiom $\Box\Diamond p \to \Diamond\Box p$ nor the Löb axiom $\Box(\Box p \to p) \to \Box p$ is globally elementary. The first case follows again by a violation of the Skolem-Löwenheim theorem ([vB76] or [vB84], see also [Gol75]). The Löb axiom defines the class of transitive frames containing no infinite R-paths. A compactness argument suffices to show that this class is not first-order definable.

The task entailed by the forgoing discussion is now clear: Try to characterize the classes of locally and globally elementary modal formulae. Whereas second-order logic poses insurmountable computational difficulties, there are well developed algorithmic tools for first-order logic, and it is generally very well studied. This makes it eminently worth while to try and identify (as large as possible) elementary fragments of modal logic.

One model theoretic characterization is the following:

Theorem 1.1.3 ([vB84]) *For any modal formula φ, the following are equivalent:*[1]

1. *φ is globally elementary*
2. *φ is preserved under ultrapowers*
3. *φ is preserved under elementary equivalence.*

This is a very neat characterization, but the constructions involved are complicated and have infinitely many instances. If possible, we would prefer a theorem that gives us a simple way to check whether a formula is elementary. This would, of course, be much too good to be true, and indeed, our skepticism is confirmed by Chagrova's theorem:

Theorem 1.1.4 ([Cha91], [CC06]) *It is algorithmically undecidable whether a given modal formula is elementary.*

An effective characterization is therefore impossible, but if we are willing to be satisfied with approximations, all is not lost. Various large and interesting, syntactically defined classes of (locally) elementary formulae are known. These are presented in section 1.3. But first we discuss another desirable property of modal formulae, often accompanying elementarity, namely that of *canonicity.*

1.2 Canonicity

Together with the elementarity of modal formulae, we will be concerned with the related property of their canonicity. We begin by quickly reviewing the canonical model construction — a general method for proving completeness of modal logics from which this property takes its name.

1.2.1 Canonical models

Definition 1.2.1 Given a modal logic Λ, a set of formulae Γ is a Λ-*maximal consistent set* (Λ-MCS) if it is Λ-consistent but every proper superset of Γ is Λ-inconsistent.

A modal version of Lindenbaum's lemma holds, saying that every Λ-consistent set of formulae is contained in a Λ-MCS.

Definition 1.2.2 The *canonical model* of a modal logic Λ is the model $\mathcal{M}_\Lambda = (W_\Lambda, R_\Lambda, V_\Lambda)$ where

1. W_Λ is the set of all Λ-MCS's,

[1]The reader unfamiliar with these constructions is referred to any standard text on model theory, e.g. [Hod93] or [CK90].

2. for all Λ-MCS's $u, v \in W_\Lambda$, let $R_\Lambda uv$ iff $\psi \in v$ implies $\Diamond\psi \in u$ for all formulae φ,

3. for every propositional variable p, let $V_\Lambda(p) = \{w \in W_\Lambda \mid p \in w\}$.

The central fact about the canonical model is that in it "truth = membership", i.e.

Lemma 1.2.3 *For any modal logic Λ, any Λ-MCS w, and any formula φ,*

$$(\mathcal{M}_\Lambda, w) \Vdash \varphi \quad \textit{iff} \quad \varphi \in w.$$

In other words, every set of Λ-consistent formulae is satisfied at some point in the canonical model. This fact is sufficient for the strong completeness of Λ, provided that $\mathcal{M}_\Lambda$ is based on a frame for Λ. In other words, to conclude that Λ is strongly complete with respect to its Kripke frames, we will first have to show that the *canonical frame* $\mathfrak{F}_\Lambda = (W_\Lambda, R_\Lambda)$ validates Λ. For example, since the minimal logic **K** is valid in *every* Krikpe frame, we immediately have:

Proposition 1.2.4 **K** *is sound and strongly complete with respect to the class of all Kripke frames.*

We will call a logic Λ *canonical* if $\mathfrak{F}_\Lambda \Vdash \Lambda$. A formula φ is called *canonical* if $\mathbf{K} \oplus \varphi$ is a canonical logic. Not every logic is canonical, particularly, by the above no incomplete logic can be canonical. (For examples of incomplete logics see [BdRV01] or [CZ97].) An example of a complete but non-canonical logic is provided by the logic obtained by adding the McKinsey axiom $\Box\Diamond p \to \Diamond\Box p$ to **K**. That this logic is complete was proved by Fine in [Fin75a]. Whether or not the McKinsey axiom is canonical was an open question for quite a number of years. It was finally shown to be non-canonical by Goldblatt in [Gol91].

The usual method of proving that a logic is canonical, considers a structure halfway between the canonical model and canonical frame, namely the *canonical general frame* $\mathfrak{g}_\Lambda$. $\mathfrak{g}_\Lambda$ augments the canonical frame with an algebra of admissible sets $\mathbb{W}_\Lambda$, consisting of all subsets of W_Λ of the form $\{w \in W_\Lambda \mid \varphi \in w\}$ for formulae $\varphi \in \mathcal{L}$. Using the fact that logics are closed under substitution it is easy to see that:

Proposition 1.2.5 $\mathfrak{g}_\Lambda \Vdash \Lambda$, *for any normal modal logic Λ, that is, every logic is valid on its canonical general frame.*

This yields the general completeness result with respect to general frames mentioned in chapter 0, namely:

Theorem 1.2.6 *Every normal modal logic is strongly complete with respect to its class of general frames.*

To prove that a logic Λ is canonical, we can try to transfer its validity in $\mathfrak{g}_\Lambda$ to $\mathfrak{F}_\Lambda$. In other words, we can try to show that the theorems of the logic exhibit a suitable kind of persistence, formally:

Proposition 1.2.7 *If a logic Λ is (strongly) complete with respect to a class of general frames $\mathfrak{C}$, then the $\mathfrak{C}$-persistence of Λ implies its (strong) completeness with respect to its class of Kripke frames.*

It can be shown that the canonical general frame of any normal modal logic is descriptive (see e.g. [BdRV01]) and hence that any d-persistent logic is canonical. Particularly, to show that a logic is canonical, it is enough to show that its axioms are d-persistent. This gives rise to the idea of a canonical formula:

Definition 1.2.8 A formula φ is called *canonical* if it is d-persistent.

Of course we could simply have defined a canonical formula as a formula φ such that the logic $\mathbf{K} \oplus \varphi$ is valid on its canonical frame. Such a definition would be problematic, however. For example, the canonical model varies depending on the cardinality of the set of propositional variables over which the language is built, or with enrichments of the language with e.g. nominals.

The canonical models associated with hybrid logics need not be descriptive, hence d-persistence is not the correct notion of canonicity for these logics. Appropriate canonicity notions for hybrid logics are treated in chapter 5.

We concluded the previous section by quoting Chagrova's theorem (theorem 1.1.4) which states the algorithmic undecidability of the elementarity problem for modal formulae. Indeed, an analogue of this theorem also holds for canonicity:

Theorem 1.2.9 *It is algorithmically undecidable whether a given modal formula is canonical.*

As will be seen in section 1.5, the above theorem can be obtained as a corollary of a theorem from [CZ93].

1.2.2 Canonicity and elementarity

Elementarity and canonicity are properties that often (but not always!) go hand-in-hand for modal formulae. A logic is *determined* by a class of frames if it is sound and complete with respect to that class. Note that this need not be the class of all its Kripke frames.

Theorem 1.2.10 ([Fin75b]) *If a modal logic Λ is determined by some elementary class of frames then it is canonical.*

For formulae the theorem can be rephrased as follows: If a modal formula φ is globally elementary, then $\mathbf{K} \oplus \varphi$ is complete (with respect to its class of Kripke frames) iff φ is canonical.

It was recently shown by Goldblatt, Hodkinson and Venema ([GHV03]) that the converse of theorem 1.2.10 does not hold.

	Modal formula	Local first-order frame correspondent
Reflexivity	$\Box p \to p$	Rxx
Transitivity	$\Box p \to \Box\Box p$	$\forall y \forall z(Rxy \wedge Ryz \to Rxz)$
Symmetry	$p \to \Box\Diamond p$	$\forall y(Rxy \to Ryx)$
Partial functionality	$\Diamond p \to \Box p$	$Rxy \wedge Rxz \to y = z.$
Seriality	$\Box p \to \Diamond p$	$\exists y Rxy$
Church-Rosser	$\Diamond\Box p \to \Box\Diamond p$	$\forall y \forall z(Rxy \wedge Rxz \to \exists u(Ryu \wedge Rzu))$
McKinsey + Transitivity	$(\Box\Diamond p \to \Diamond\Box p) \wedge (\Box p \to \Box\Box p)$	$\forall y \forall z(Rxy \wedge Ryz \to Rxz) \wedge \exists y(Rxy \wedge \forall z(Ryz \to z = y))$

Table 1.1: Some well known correspondences

Formula	Globally Elementary	Canonical	Strongly Complete	Weakly Complete
$\Box p \to p$	Yes, [Sah75]	Yes[Sah75]	Yes	Yes
$\Box\Diamond\top \to \Box(\Box(\Box p \to p) \to p)$	Yes	No	No	No [vB83]
$\Diamond\Box p \to (\Diamond\Box(p \wedge q) \vee \Diamond\Box(p \wedge \neg q))$	No, [Fin75b]	Yes, [Fin75b]	Yes	Yes
$\Box(\Box p \to p) \to \Box p$ (Löb-axiom)	No	No	No [BdRV01]	Yes (Segerberg, see [BdRV01])
$\Box\Diamond p \to \Diamond\Box p$	No [vB83]	No [Gol91]	?	Yes [Fin75a]
$\Box(\Box p \leftrightarrow p) \to \Box p$	No[BS85]	No	No	No[BS85]

Table 1.2: Elementarity, completeness and canonicity

1.3 Syntactic classes

We concluded section 1.1 by remarking that, in the light of Chagrova's theorem, we will have to content ourselves with decidable *approximations* of the class of elementary modal formulae. In section 1.2 we introduced the notion of canonicity for formulae and logics. As will be seen in subsection 1.5.1, both the class of canonical formulae and the class of elementary and canonical formulae are also undecidable. So here too decidable approximations will have to suffice.

The best known such approximations take the form of syntactically specified classes of formulae with accompanying correspondence and canonicity results. In this section we review the most famous among these classes and establish some relationships between them.

1.3.1 Shallow formulae

The modal depth of a formula is the maximum depth of nesting of modal operators in the formula. More formally:

Definition 1.3.1 Given a formula $\varphi \in \mathcal{L}_\tau$, we define the *modal depth of* φ, denoted $depth(\varphi)$, inductively:

- $depth(p) = depth(\bot) = depth(\top) = 0$;
- $depth(\neg\varphi) = depth(\varphi)$;
- $depth(\varphi\wedge\psi) = depth(\varphi\vee\psi) = depth(\varphi \to \psi) = depth(\varphi \leftrightarrow \psi) = \max\{depth(\varphi), depth(\psi)\}$;
- $depth(\langle\varphi\rangle) = depth(\Box\varphi) = 1 + depth(\varphi)$.

The following proposition from [vB83] captures what is probably the simplest class of elementary and canonical formulae:

Proposition 1.3.2 ([vB83], lemma 9.7) *Every modal formula of depth at most 1 is locally elementary.*

Ten Cate ([tC05b]) defines a slightly more general class, viz. the *shallow formulae* are those formulae in which every occurrence of a propositional variable is in the scope of at most one modal operator. He goes on to show that all shallow formulae are locally persistent with respect to the class of refined frames. Now (local) persistence with respect to refined frames implies (local) elementarity. This can be seen from the fact that all elementary general frames are refined, and hence that formulae persistent with respect to these general frames are an Benthem formulae (see subsection 1.3.4, below). Moreover, every descriptive frames is refined, and hence formulae persistent with respect to refined frames are canonical. Combining this fact we have:

Proposition 1.3.3 *All shallow formulae are locally elementary and canonical.*

1.3.2 Sahlqvist formulae and Sahlqvist–van Benthem formulae

The Sahlqvist formulae undoubtedly form the best known syntactically specified class of elementary and canonical formulae. They were first introduced in [Sah75], in a slightly more restricted form that the present definition. Fix an arbitrary monadic similarity type κ for the rest of this subsection. All definitions in this section may be extended to reversive languages by simply treating inverse diamonds and boxes exactly like their non-inverted counterparts.

Definition 1.3.4 An occurrence of a propositional variable or nominal in a formula φ is *positive* (*negative*) if it is in the scope of an even (odd) number of negation signs. A formula φ is *positive (negative) in propositional variable* p if all occurrences of p in φ are positive (negative). A formula is *positive (negative)* if it is positive (negative) in all propositional variables. The positivity or negativity of a formula in a propositional variable will be referred to its *polarity* in that propositional variable.

Definition 1.3.5 A *boxed atom* is a propositional variable, prefixed with finitely many (possibly none) boxes, i.e., a formula of the form $\Box_{\alpha_1} \dots \Box_{\alpha_n} p$ for some $n \in \omega$ and $\alpha_1, \dots, \alpha_n \in \kappa$. A *Sahlqvist antecedent* is a formula built up from $\top$, $\bot$, boxed atoms, and negative formulae, using $\wedge$, $\vee$ and diamonds. A *Sahlqvist implication* is a formula of the form $\varphi \to \mathsf{Pos}$ where φ is a Sahlqvist antecedent and Pos is a positive formula. In particular, note that any negative formula is a Sahlqvist antecedent. A *Sahlqvist formula* is built up from Sahlqvist implications by applying conjunctions, disjunctions, and boxes.

[BdRV01] define subclasses of the Sahlqvist formulae, called the *very simple* and the *simple Sahlqvist formulae*, as follows.

Definition 1.3.6 A *very simple Sahlqvist antecedent* is any formula constructed from $\top$, $\bot$, and propositional variables by applying $\wedge$ and diamonds. A *very simple Sahlqvist formula* is an implication with a very simple Sahlqvist antecedent as antecedent, and a positive formula as consequent.

A *simple Sahlqvist antecedent* is any formula constructed from $\top$, $\bot$, and boxed atoms by applying $\wedge$ and diamonds. A *simple Sahlqvist formula* is an implication with a simple Sahlqvist antecedent as antecedent, and a positive formula as consequent.

Many well known formulae fall within the class of Sahlqvist formulae, for example: the axioms for reflexivity ($p \to \Diamond p$), transitivity ($\Diamond\Diamond p \to \Diamond p$ or $\Box p \to \Box\Box p$), symmetry ($p \to \Box\Diamond p$), seriality ($\Box p \to \Diamond p$), and the Geach-formula ($\Diamond\Box p \to \Box\Diamond p$, defining the Church-Rosser property). Among these formulae, the reflexivity and symmetry axioms as well as the diamond-version of the transitivity axiom are moreover very simple Sahlqvist formulae. The McKinsey axiom3 ($\Box\Diamond p \to \Diamond\Box p$) is not a Sahlqvist formula — the diamond in the scope of a box in the antecedent rules it out — nor, being not first-order definable (see [vB76]), can it be frame equivalent to a Sahlqvist formula. Examples of elementary and canonical formulae which are not equivalent to any Sahlqvist formula, will be discussed later in this section.

Lemma 1.3.7 *Let φ be a Sahlqvist formula, and φ' the formula obtained from $\neg\varphi$ by importing the negation over all connectives. Then φ' is a Sahlqvist antecedent.*

Proof. By induction on the construction of φ from Sahlqvist implications. If φ is a Sahlqvist implication $\alpha \to \mathsf{Pos}$, negating and rewriting it as $\alpha \wedge \neg\mathsf{Pos}$ already turns it into a Sahlqvist antecedent. If $\varphi = \Box\psi$, where ψ satisfies the claim, then $\neg\varphi \equiv \Diamond\neg\psi$ hence the claim follows for φ, because Sahlqvist antecedents are closed under diamonds. Likewise, if $\varphi = \psi_1 \wedge \psi_2$, where ψ_1 and ψ_2 satisfy the claim, then $\neg\varphi \equiv \neg\psi_1 \vee \neg\psi_2$ hence the claim follows for φ, because Sahlqvist antecedents are closed under disjunctions. The case of $\varphi = \psi_1 \vee \psi_2$ is completely analogous. QED

We now have the following proposition:

Proposition 1.3.8 *Every Sahlqvist formula is semantically equivalent to a negated Sahlqvist antecedent, and hence to a Sahlqvist implication.*

The usual definition of the Sahlqvist formulae (see for example [BdRV01]) differs from definition 1.3.5 in that disjunctions are only allowed between formula that share no propositional variables. This would exclude a formula like $(\Diamond\Diamond p \to \Diamond p) \vee (\Diamond\Box p \to \Box\Diamond p)$, which would be admitted by our definition.

In the light of proposition 1.3.8 it is clear that this restriction on the occurrence of disjunctions is unnecessary as far as the elementarity and canonicity of the members of the obtained classes are concerned.

However, this requirement is indeed essential for the usual proof of elementarity to succeed (see once again e.g. [BdRV01]), which treats the main disjuncts of the formula separately, finding a frame correspondent for each, and then taking the disjunction of these as a frame correspondent for the formula as a whole. Now in general, if A and B are frame correspondents for φ and ψ, respectively, $A \vee B$ need not be a frame correspondent for $\varphi \vee \psi$ unless $\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi) = \emptyset$. Indeed, $x \neq x$ is a local frame correspondent for both $\neg p$ and $\Diamond p$, while $(x \neq x) \vee (x \neq x)$ does not correspond to $\neg p \vee \Diamond p$, which corresponds to reflexivity.

In [dRV95] de Rijke and Venema define an extended class of Sahlqvist formulae which allow for the unrestricted use polyadic diamonds in the antecedents. We will not discuss this class further here. It is subsumed by the polyadic inductive formulae of Goranko and Vakarelov, to be introduced in section 4.2.

Another natural syntactic generalization of the class of Sahlqvist formulae is provided in [vB83]. Following [Kra99] we will refer to this extended class as the class of *Sahlqvist–van Benthem formulae*. It is defined as follows:

Definition 1.3.9 A *Sahlqvist–van Benthem formula* is an $\mathcal{L}_\kappa$-formula in negation normal form, such that for every propositional variable p, either

(SvB-Pos) there is no positive occurrence of p in a subformula $\psi \wedge \chi$ or $\Box_\alpha\psi$ which is in the scope of a $\Diamond_\beta$, or

(SvB-Neg) there is no negative occurrence of p in a subformula $\psi \wedge \chi$ or $\Box_\alpha\psi$ which is in the scope of a $\Diamond_\beta$.

Below we will need the following dual version of this definition.

Definition 1.3.10 A formula φ written in negation normal form is a *dual Sahlqvist–van Benthem formula* if, for every propositional variable p, it satisfies either

(DSvB-Pos) there is no positive occurrence of p in a subformula $\psi \vee \chi$ or $\Diamond_\alpha \psi$ which is in the scope of a $\Box_\beta$, or

(DSvB-Neg) there is no negative occurrence of p in a subformula $\psi \vee \chi$ or $\Diamond_\alpha \psi$ which is in the scope of a $\Box_\beta$.

Hence, a formula φ is the negation of a Sahlqvist–van Benthem formula rewritten in negation normal form, if and only if it is a dual Sahlqvist–van Benthem formula.

It should be clear that all Sahlqvist formulae, after being rewritten in negation normal form, are Sahlqvist–van Benthem. In particular, condition (SvB-Neg) is satisfied with respect to every propositional variable in a Sahlqvist formula.

The example $\Diamond(p \wedge \Box\Diamond\neg p) \to (\Diamond\Box p \vee \Box\Box\neg p)$ is used in [Kra99] to show that not every Sahlqvist–van Benthem formula is Sahlqvist. It is however not difficult to rewrite this formula as a Sahlqvist formula whilst maintaining semantic equivalence, namely as the formula $[\Diamond(p \wedge \Box\Diamond\neg p) \wedge \Box\Diamond\neg p \wedge \Diamond\Diamond p] \to \bot$. Had the polarity of p been reversed, this would not have succeeded however — to obtain a Sahlqvist formula we would have had to switch the polarity. We have the following proposition:

Proposition 1.3.11 *Every Sahlqvist–van Benthem formula is locally equivalent to a Sahlqvist implication.*

Proof. Let φ be a Sahlqvist–van Benthem formula. Obtain φ_1 from φ by switching the polarity of each occurring variable which does not satisfy condition (SvB-Neg) of definition 1.3.9. This maintains local equivalence. Let φ_2 be obtained by rewriting $\neg\varphi_1$ in negation normal form. Hence φ_2 is a dual Sahlqvist–van Benthem formula in which each occurring propositional variable satisfies condition (DSvB-Pos) of definition 1.3.10. It follows that in φ_2, whenever a positive variable occurrence is in the scope of a box occurrence, the only other connectives in the scope of that box occurrence are conjunctions and boxes. Let φ_3 be obtained from φ_2 by distributing boxes over conjunctions as much as possible. Then φ_3 is a Sahlqvist antecedent. Hence $\varphi_3 \to \bot$ is a Sahlqvist implication, locally equivalent to φ. QED

1.3.3 Inductive formulae

The inductive formulae were introduced and studied by Goranko and Vakarelov in [GV01], [GV02] and [GV06]. In the first two of these papers, these formulae were referred to as *polyadic Sahlqvist formulae*. We will use the terminology of the third paper, i.e., *inductive formulae*. Here we will only define the so-called *monadic inductive formulae*, postponing the definition of the full fragment until chapter 4, where polyadic languages are treated in detail. Fix a monadic modal similarity type κ for the rest of this subsection.

Definition 1.3.12 Let $\sharp$ be a symbol not belonging to $\mathcal{L}_\kappa$. Then a *box-form of* $\sharp$ in $\mathcal{L}_\kappa$ is defined recursively as follows:

1. $\sharp$ is a box-form of $\sharp$;
2. If $\mathbf{B}(\sharp)$ is a box-form of $\sharp$ and $\alpha \in \kappa$, then $\Box_\alpha \mathbf{B}(\sharp)$ is a box-form of $\sharp$.
3. If $\mathbf{B}(\sharp)$ is a box-form of $\sharp$ and A is a positive formula, then $A \to \mathbf{B}(\sharp)$ is a box-form of $\sharp$.

Thus, box-forms of $\sharp$ are, up to semantic equivalence, of the type

$$A_0 \to \Box_1(A_1 \to \ldots \Box_n(A_n \to \sharp)\ldots)$$

where $\Box_1, \ldots, \Box_n$ are sequences of boxes in $\mathcal{L}_\kappa$ and $A_1, \ldots, A_n$ are positive formulae (possibly, just $\top$).

Definition 1.3.13 By substituting a propositional variable p for $\sharp$ in a box-form $\mathbf{B}(\sharp)$ we obtain a *box-formula of* p, namely $\mathbf{B}(p)$. The last occurrence of the variable p is the *head* of $\mathbf{B}(p)$ and every other occurrence of a variable in $\mathbf{B}(p)$ is *inessential* there.

Definition 1.3.14 A *monadic regular formula* is a modal formula built up from $\top$, $\bot$, positive formulae and negated box-formulae by applying conjunctions, disjunctions, and boxes.

Definition 1.3.15 The *dependency digraph* of a set of box-formulae $\mathcal{B} = \{\mathbf{B}_1(p_1), \ldots, \mathbf{B}_n(p_n)\}$ is a digraph $G_\mathcal{B} = \langle V, E\rangle$ where $V = \{p_1, \ldots, p_n\}$ is the set of heads in $\mathcal{B}$, and the edge set E is such that $p_i E p_j$ iff p_i occurs as an inessential variable in a box-formula from $\mathcal{B}$ with a head p_j. A digraph is *acyclic* if it does not contain oriented cycles or loops.

We will also talk about the dependency digraph of a formula, when we mean the dependency digraph of the set of box-formulae that occur as subformulae of the formula.

Definition 1.3.16 A *monadic inductive formula (MIF)* is a monadic regular formula with an acyclic dependency digraph.

Example 1.3.17 An example of a monadic inductive formula, which is not a Sahlqvist or Sahlqvist–van Benthem formula, is the formula

$$D = p \land \Box(\Diamond p \to \Box q) \to \Diamond\Box\Box q \equiv_{\text{sem}} \neg p \lor \neg\Box(\Diamond p \to \Box q) \lor \Diamond\Box\Box q,$$

obtained as a disjunction of the negated box-formulae $\neg p$ and $\neg\Box(\Diamond p \to \Box q)$, and the positive formula $\Diamond\Box\Box q$. The dependency digraph of D over the set of heads $\{p, q\}$ has only one edge, from p to q.

An example of a regular, but non-inductive formula is $\Box((\neg\Box p \lor q) \lor (\neg q \lor \Box p)) \lor \neg p$, because the heads p and q depend on each other. $\lhd$

So we have yet another syntactically defined class of elementary and canonical formulae. The question is now, how does this class compare to the previously defined classes? Has this more complicated definition gained us any ground? The following propositions provide some answers.

Proposition 1.3.18 *Every Sahlqvist formula is semantically equivalent to a monadic inductive formula.*

Proof. We note that, up to semantic equivalence, we could have equivalently defined the Sahlqvist formulae as the class of all formulae built up from negated boxed atoms and positive formulae, using conjunction, disjunction, and boxes. But this is exactly the definition of the regular formulae with 'box formulae' replaced with 'boxed atoms'. Since every boxed atom is a box formula, we see that every Sahlqvist formula is semantically equivalent to a regular formula. Moreover, the dependency digraph of any Sahlqvist formula, seen thus as a regular formula, contains no edges. The proposition follows. QED

From propositions 1.3.11 and 1.3.18 we have the following corollary:

Corollary 1.3.19 *Every Sahlqvist–van Benthem formula is locally equivalent to a monadic inductive formula.*

What about the converse? The following proposition is proved in [GV06]. The proof makes use of special general frames called *ample*, and of the accompanying notion of persistence with respect to these general frames, viz. *a-persistence*. It is shown that every Sahlqvist formula is locally a-persistent, but that the formula D form example 1.3.17 fails to be a-persistent.

Proposition 1.3.20 ([GV06]) *The formula* $D = p \wedge \Box(\Diamond p \to \Box q) \to \Diamond\Box\Box q$ *is not frame equivalent to any Sahlqvist formula.*

Corollary 1.3.21 *The formula D is not frame equivalent, and hence not locally frame equivalent, to any Sahlqvist–van Benthem formula.*

To summarize — every (local) property of Kripke frames definable by a Sahlqvist or Sahlqvist–van Benthem formulae is definable by a monadic inductive formula. Moreover there are monadic inductive formulae which define (local) properties of Kripke frames which are not definable by any Sahlqvist or Sahlqvist–van Benthem formula. In other words, the class of monadic inductive formulae is indeed a strict superclass of the Sahlqvist and Sahlqvist–van Benthem formulae, even when considered modulo frame equivalence.

1.3.4 Van Benthem-formulae

In the preceding subsections we have introduced a hierarchy of three classes, starting with the Sahlqvist formulae, moving up to the Sahlqvist–van Benthem formulae and then on to the monadic inductive formulae. Each class subsumed its predecessor. We now place the cappingstone on this hierarchy. The following class, which has been dubbed 'van Benthem-formulae' in [CGV05], was introduced in [vB83].

Definition 1.3.22 The class of *van Benthem-formulae* consists of all $\mathcal{L}$-formulae φ such that $\mathrm{ST}(\varphi, x)$ is logically implied by (and hence logically equivalent to) the set of L_0 substitution instances[2] of L_1-formulae logically equivalent to $\mathrm{ST}(\varphi, x)$. The *global van Benthem-formulae* are defined similarly by considering $\forall x \mathrm{ST}(\varphi, x)$.

[2] There are some technicalities regarding the handling of the variables involved in this substitution which are irrelevant for our purposes and which we omit for the sake of brevity. The interested reader may consult definition 9.11 in [vB83].

That the classes of Sahlqvist, Sahlqvist–van Benthem and inductive formulae are subsumed by this definition is seen by inspecting the respective proofs of the elementarity of the members of these classes. Indeed, each proceeds by computing the first-order definable 'minimal valuations' of the propositional variables and substituting these. To use the terminology of [vB83], each of these classes is amenable to the *method of substitutions*. The van Benthem-formulae constitute *exactly* the class of formulae for which first-order correspondences may be established by this method. Here is a simple example.

Example 1.3.23 We consider again the transitivity axiom $\Box p \to \Box\Box p$. The standard translation yields $\forall y(Rxy \to P(y)) \to \forall u(Rxu \to \forall v(Ruv \to P(v)))$. The smallest valuation / interpretation of P that will make the antecedent true consists of the set of all successors of x, viz. $\lambda w.Rxw$. Substituting this for P we obtain $\forall y(Rxy \to Rxy) \to \forall u(Rxu \to \forall v(Ruv \to Rxv))$. The antecedent becomes valid yielding $\forall u(Rxu \to \forall v(Ruv \to Rxv))$, defining the 'local transitivity' of the current state, as expected. $\lhd$

By their very definition all van Benthem-formulae are locally elementary. However, not all of them are canonical, as is witnessed by the formula $\Box\Diamond\top \to \Box(\Box(\Box p \to p) \to p)$ from [vB83], which in locally equivalent to its substitution instance $\Box\Diamond\top \to \Box(\Box(\Box\bot \to \bot) \to \bot)$ which reduces to $\Box\Diamond\top \to \Box\bot$, but which axiomatizes an incomplete logic.

An alternative characterization of the van Benthem-formulae can be given in terms of persistence.

Definition 1.3.24 A general frame $\mathfrak{g} = (\mathfrak{F}, \mathbb{W})$ is *elementary* if, for every L_0-formula $\alpha(x, x_1, \ldots, x_n)$ and all points $w_1, \ldots, w_n$ the set

$$\{w \in \mathfrak{F} \mid \mathfrak{F} \models \alpha[x := w, x_1 := w_1, \ldots, x_n := w_n]\}$$

is admissible. In other words, a general frame is elementary if all subsets of its domain definable with parameterized L_0-formulae are admissible.

A formula is called (locally) *e-persistent* if it is (locally) persistent with respect to the class of all elementary general frames. In [vB83] it is shown that the locally e-persistent formulae are exactly the van Benthem-formulae. (Strictly speaking, only the inclusion of the e-persistent formulae in the van Benthem-formulae is shown in [vB83]. A proof of the other inclusion can be found in [GO06].) Formulated as a proposition:

Proposition 1.3.25 *A formula $\varphi \in \mathcal{L}$ is a van Benthem-formula iff it is locally e-persistent.*

The van Benthem-formulae form a recursively enumerable ([vB83]) but, as we will see below in subsection 1.5.1, undecidable set.

1.3.5 Modal reduction principles over transitive frames

A *modal reduction principle* is an $\mathcal{L}$-formula of the form $Q_1Q_2 \ldots Q_np \to Q_{n+1}Q_{n+2} \ldots Q_{n+m}p$ where $0 \leq n, m$ and $Q_i \in \{\Box, \Diamond\}$ for $1 \leq i \leq n+m$. Many well known modal axioms take this form, e.g. $\Box p \to p$ (reflexivity), $\Box p \to \Box\Box p$ (transitivity), $p \to \Box\Diamond p$ (symmetry),

$\Diamond\Box p \to \Box\Diamond p$ (the Geach axiom) and the $\Box\Diamond p \to \Diamond\Box p$ (the McKinsey axiom). In [vB76] van Benthem provides a complete classification of the modal reduction principles that define first-order properties on frames. For example, as we have already seen, $\Diamond\Box p \to \Box\Diamond p$ defines such a property and $\Box\Diamond p \to \Diamond\Box p$ does not.

It is also shown that, however, when we restrict attention to transitive frames, *all* modal reduction principles define first-order properties. Or formulated differently, every formula of the form $(\Box p \to \Box\Box p) \wedge \psi$ with ψ a modal reduction principle has a first-order global frame correspondent.

What about canonicity? That $(\Box p \to \Box\Box p) \wedge (\Box\Diamond p \to \Diamond\Box p)$, i.e. the McKinsey axiom together with transitivity, is canonical was already proven by Lemmon in [Lem77]. In [Jón94] Jónsson gives an algebraic proof of this fact.

In [Zak97a] Zakharyaschev proves that any extension of **K4**[3] axiomatized with modal reduction principles has the finite model property, that is to say, any non-theorem of such a logic is refuted on a finite model based on a frame for that logic. Combining this fact with the elementarity of the reduction principles over transitive frames, the next proposition now follows by theorem 1.2.10.

Proposition 1.3.26 *Every $\mathcal{L}$-formula of the form $(\Box p \to \Box\Box p) \wedge \psi$ with ψ a modal reduction principle is globally elementary and canonical.*

This result is interesting in that it does not fit into the Sahlqvist-Inductive hierarchy above, and not only because of syntactic considerations. It is shown in [vB83] that

Proposition 1.3.27 ([vB83]) *The formula $(\Box p \to \Box\Box p) \wedge \Box\Diamond p \to \Diamond\Box p$ is not a global van Benthem formula, nor is $(\Box p \to \Box\Box p) \wedge \Box(\Box p \to \Box\Box p) \wedge \Box\Diamond p \to \Diamond\Box p$ a (local) van Benthem formula.*

The formulae mentioned in proposition 1.3.27 also prove to be the nemesis of all currently known algorithmic approaches to correspondence and canonicity, which approaches are the main concern of this thesis. That these formulae would be such hard nuts to crack could perhaps be anticipated form the fact that the proofs of their elementarity require a version of the axiom of choice.

1.3.6 Complex formulae

To conclude our list of syntactic classes of elementary and canonical formulae, we briefly mention Vakarelov's *complex Sahlqvist formulae*, introduced in [Vak03a]. See also [Vak03b] and [Vak05].

These formulae can be seen as substitution instances of Sahlqvist formulae obtained through the substitution of certain elementary disjunctions for propositional variables. The resulting formulae violate the Sahlqvist definition, as the antecedents contain disjunctions of propositional variables and their negations in the scope of boxes.

Every complex formula is locally equivalent to a Sahlqvist formulae which may be obtained from it through a suitable substitution. This substitution is effectively computable,

[3] **K4** is the logic $\mathbf{K} \oplus \Box p \to \Box\Box p$, which is determined by the class of all transitive frames.

but involves some non-trivial combinatorics, and often requires the introduction of a *strictly greater* number of propositional variables. It follows that every complex formula is locally elementary and canonical.

Example 1.3.28 This example is taken from [Vak03a]. The formula

$$\Diamond\Box(p \vee q) \wedge \Diamond\Box(p \vee \neg q) \wedge \Diamond\Box(\neg p \vee q) \to \Box\Diamond(p \wedge q)$$

is a complex formula, but not a Sahlqvist formula. ◁

1.4 Algorithmic classes

The proof of the elementarity of the Sahlqvist formulae may be given in the form of an algorithm based upon the method of substitutions — the so-called *Sahlqvist–van Benthem algorithm* (see e.g. [BdRV01]). However, the syntactic definition is taken as primary as far as this class (and all other syntactically specified classes) is concerned. But there are good reasons why one might like to turn the tables and rather take the *algorithm* as primary.

It has often been remarked that the syntactic approach to the delineation of classes of elementary and canonical formulae has reached is practical limits. In [Gol91] Glodblatt remarks that

> "the main result of the present paper [viz. the non-canonicity of the McKinsey axiom] indicates that there is no natural way to extend Sahlqvist's scheme to obtain a larger class of canonical formulae".

Weighing the advantages of syntactic extensions of the Sahlqvist class, Blackburn et. al. say ([BdRV01], p. 169):

> "By adding further restrictions it is possible to extend it [the Sahlqvist class] further, but it is not obvious that the resulting loss of simplicity is really worth it."

Now, depending on what one considers to be 'simple' and 'natural', the inductive formulae might be construed as either justifying these misgiving or as proving them incorrect, or at least premature. However that may be, the point is still well taken — syntactically, things can probably become only increasingly convoluted from here onwards.

These consideration serve as motivation for what we term the *algorithmic* approach to the delineation of classes of elementary and canonical formulae. Since we are interested in *decidable* classes of elementary and canonical formulae, why not define such classes in terms of the algorithms (decision procedures) which justify their decidability? This is exactly what we will do in this thesis.

1.4.1 Second-order quantifier elimination

The second-order quantifier elimination problem asks the question whether a given a second-order formula is equivalent to a formula containing no second-order quantifiers, i.e. whether it is equivalent to a first-order formula. A classical reference on this problem is [Ack35]. For the current state of the art, the reader is referred to [GSS06].

By considering the standard second-order translations $\forall\overline{P}\mathrm{ST}(\varphi, x)$ of a modal formula φ, it is clear that the elementarity problem for modal formulae can be seen as essentially a second-order quantifier elimination problem. With the help of theorem 1.1.4, this is one way of seeing that second-order quantifier elimination is undecidable.

Some approaches to the second-order quantifier elimination problem take the form of (partial) algorithms, which attempt to reduce second-order input-formulae to equivalent first-order formulae. The best-known such algorithms are probably SCAN and DLS, which we will discuss below.

Our interest in these algorithms of course lies in the fact that, by the above remarks, any second-order quantifier elimination algorithm may be employed to try and find first-order (local) frame correspondents for modal formulae.

1.4.2 SCAN

The algorithm SCAN was introduced in [GO92]. SCAN accepts existential second-order formulae as input which it attempts to reduce equivalently to first-order formulae as follows:

Clausal form The input is transformed into clausal form. All existential individual quantifiers are skolemized.

Constraint resolution The algorithm now tries to deduce an equivalent of the (clausified) input, containing no second-order variables. This is done by generating sufficiently many logical consequences of the input, and keeping only those that contain no second-order variables. To accomplish this task, a special resolution calculus called *constraint resolution*, or c-resolution, is used. The most significant rules of the c-resolution calculus are:

C-resolution: $$\frac{C \vee Q(s_1, \ldots, s_n) \quad D \vee \neg Q(t_1, \ldots, t_n)}{C \vee D \vee s_1 \not\approx t_1 \vee \ldots \vee s_n \not\approx t_n}$$

provided the two premises have no variables in common and $C \vee Q(s_1, \ldots, s_n)$ and $D \vee \neg Q(t_1, \ldots, t_n)$ are distinct clauses. The conclusion is called a *C-resolvent* with respect to Q.

C-factoring: $$\frac{C \vee Q(s_1, \ldots, s_n) \vee Q(t_1, \ldots, t_n)}{C \vee Q(s_1, \ldots, s_n) \vee s_1 \not\approx t_1 \vee \ldots \vee s_n \not\approx t_n}$$

The conclusion is called a *C-factor* with respect to Q.

Purity deletion: $$\frac{N \uplus \{C \vee Q(s_1, \ldots, s_n)\}}{N}$$

if all inferences with respect to Q with $C \vee Q(s_1, \ldots, s_n)$ as a premise have been performed.

Unskolemization Attempt to remove the introduced Skolem functions, if possible. If this fails, the algorithm might still be able to compute an equivalent without second-order quantification, but which contains Henkin-quantifiers (see e.g. [Wal70]).

The algorithm might fail for one of two reasons. Firstly, it is possible that the constraint resolution stage does not terminate due to looping. Secondly, the unskolemization might fail. Of course we should expect the algorithm to fail in some cases — some second-order formulae simply do not have first-order equivalents, even with Henkin-quantifiers. But, as one should also expect (due to undecidability, e.g. Chagrova's theorem), the algorithm also sometimes fails on formulae that have first-order correspondents, e.g. on (the translation of) the conjunction of the McKinsey and transitivity axioms from example 1.3.27. It does, however, succeed in returning an equivalent with Henkin-quantifiers for this formula! It has been shown that SCAN is powerful enough to handle all Sahlqvist formulae:

Theorem 1.4.1 ([GHSV04]) *SCAN successfully computes a first-order frame correspondent for every Sahlqvist formula.*

SCAN has been implemented and is available online. For more details on the algorithm and on the implementation see [Eng96].

1.4.3 DLS

The DLS-algorithm was formally introduced in [DŁS97]. Like SCAN, this algorithm accepts existential second-order formulae as input. It attempts to eliminate the predicate variables through the application of the equivalences of Ackermann's lemma:

Lemma 1.4.2 (Ackermann's Lemma, [Ack35]) *Let P be an n-ary predicate variable and $A(\overline{z}, \overline{x})$ a first-order formula not containing P. Then, if, $B(P)$ is negative in P, the equivalence*

$$\exists P \forall \overline{x}((\neg A(\overline{z}, \overline{x}) \vee P(\overline{x})) \wedge B(P)) \equiv B[A(\overline{z}, \overline{x})/P]$$

holds, with $B[A(\overline{z}, \overline{x})/P]$ the formula obtained by substituting $A(\overline{z}, \overline{x})$ for all occurrences P in B, the actual arguments of each occurrence of P being substituted for $\overline{x}$ in $A(\overline{z}, \overline{x})$ every time. If $B(P)$ is positive in P, then the following equivalence holds:

$$\exists P \forall \overline{x}((\neg P(\overline{x}) \vee A(\overline{z}, \overline{x})) \wedge B(P)) \equiv B[A(\overline{z}, \overline{x})/P].$$

The algorithm consists of the following four phases

Phase 1: Preprocessing This phase attempts to separate positive and negative occurrences of the predicate variable P, chosen for elimination, by transforming the input formula $\exists P A$ into the form

$$\exists \overline{x} \exists P[(A_1(P) \wedge B_1(P)) \vee \cdots \vee (A_n(P) \wedge B_n(P))],$$

where each $A_i(P)$ (respectively, $B_i(P)$) is positive (respectively, negative) in P. It is not always possible to obtain this form, and the algorithm may terminate and report failure at this stage.

Phase 2: Preparation for Ackermann's lemma Now the formula obtained in phase 1 is transformed into one of the forms suitable for the application of Ackermann's lemma, above. Both forms can always be obtained, but may require the introduction of Skolem functions.

Phase 3: Application of Ackermann's lemma Ackermann's lemma is applied and P is eliminated, but the resulting formula may contain Skolem functions. If these cannot be eliminated, the algorithm terminates with failure.

Phase 4: Simplification This stage applies some simplifying equivalences.

In order to eliminate multiple predicate variables, the algorithm is iterated. For varied examples of the execution of DLS on the translations of modal formulae see [Sza93] and [Sza02]. This is similar to the SQEMA-algorithm which will be introduced and studied in subsequent chapters, in that both are based on (versions of) Ackermann's lemma. We undertake a detailed investigation of DLS in chapter 3. DLS has been implemented and is available online, for details see [Gus96].

1.5 On the closure of syntactic classes under equivalence

As already indicated, the whole endeavour we call 'algorithmic correspondence and completeness' can be seen as a response to Chagrova's theorem ([Cha91], [CC06]), stating the undecidability of the elementarity problem for modal formulae.

If one aims at enlarging the decidable approximations of the class of elementary and/or canonical modal formulae, one route immediately suggests itself: what if we close the known syntactic classes under equivalences that preserve their desirable properties? Will the obtained closure classes be decidable? Some answers to this question are given in this section.

In chapter 0 we introduced a hierarchy of six equivalence notions, namely semantic, model, local, axiomatic, local frame, and frame equivalence. The class of locally elementary modal formulae is closed under each of the three local equivalence notions, while the class of globally elementary modal formulae is closed under all six equivalences. The class of canonical formulae is closed under the first four equivalences.

We will see that closure under axiomatic and frame equivalence leads to undecidability. On the positive side we show that we may close the classes of Sahlqvist and inductive formulae under semantic equivalence, whilst maintaining decidability.

1.5.1 Some undecidable cases

As it turns out, the actual picture is even darker than that painted by Chagrova's theorem. In [CZ95] Chagrov and Zakharyaschev construct a certain Sahlqvist formula F and, using the method of simulating Minsky machines with modal formulae presented in [CZ93], prove the following theorem:

Theorem 1.5.1 ([CZ93]) *The class of Kripke incomplete and non-elementary calculi above* **S4** *and the class of calculi equivalent to* $\mathbf{S4} \oplus F$ *are recursively inseparable.*

Recall that a *calculus* is a finitely axiomatizable logic, represented by finite set of axioms for it. Two calculi are *equivalent* if they have the same theorems. Recall also that **S4** is the logic axiomatizable with the Sahlqvist axioms $p \to \Diamond p$ and $\Diamond\Diamond p \to \Diamond p$, and is the logic of the class of all reflexive and transitive frames. A logic (or calculus) above **S4** is a logic containing, as theorems, all theorems of **S4**. A logic (or calculus) is *elementary* if the class of all frames for it is elementary. Lastly, two sets X and Y are *recursively inseparable* if there exists no recursive (i.e., decidable) set U such that $X \subseteq U$ and $U \cap Y = \emptyset$, or vice versa. Theorem 1.5.1 is a very strong result — witness the following corollaries:

Corollary 1.5.2 *The classes obtained by closing the class of Sahlqvist formulae (respectively, Sahlqvist–van Benthem formulae, respectively monadic inductive formulae) under frame equivalence or axiomatic equivalence, are undecidable.*

Proof. We treat the case of the class of formulae obtained by closing the inductive formulae under frame equivalence. The other cases are similar. Suppose, by way of contradiction, that this class were decidable. Let Ax be the conjunction of the axioms of a calculus above **S4**. Then, by assumption, we can effectively decide whether Ax is frame equivalent to an inductive formula or not. It follows that the class of calculi above **S4** with axioms which are frame equivalent to inductive formulae is decidable. But this is a class of elementary calculi containing those calculi equivalent to $\mathbf{S4} \oplus F$, yielding a contradiction with theorem 1.5.1. QED

Corollary 1.5.3 *It is undecidable whether a given $\mathcal{L}$-formula is d-persistent (canonical) or not.*

Proof. If we suppose the contrary, it would follow that the class of all calculi above **S4** which have d-persistent axioms is decidable, yielding a contradiction with theorem 1.5.1 QED

Yet another corollary shows that the class of van Benthem-formulae which, as we have seen, is recursively enumerable, is not decidable:

Corollary 1.5.4 *The class of van-Benthem formulae is undecidable.*

Proof. Suppose, to the contrary, that the question whether a given formulae $\varphi \in \mathcal{L}$ is a van-Benthem formula is decidable. We begin by noting that, for any two formulae $\varphi, \psi \in \mathcal{L}$, if $\varphi \equiv_{\mathrm{ax}} \psi$, then φ is a van Benthem-formula iff ψ is such a formula. (This easiest way to see this is by considering the characterization of the van Benthem-formulae as the e-persistent formulae.)

Let Ax be the conjunction of the axioms of a calculus above **S4**. Then, by the above, we can effectively decide whether Ax is axiomatically equivalent to a van Benthem-formula or not. It follows that the class of calculi above **S4** which are axiomatizable with van Benthem formulae is decidable. But this is a class of elementary calculi containing those equivalent to $\mathbf{S4} \oplus F$, once again yielding a contradiction with theorem 1.5.1. QED

Similar results on the undecidability of properties of logics can be obtained using what is known as *Thomason's Trick* (see e.g. section 9.6 in [Kra99]). This trick uses the fusion of logics to reduce the undecidable consistency problem for bi-modal logics to the problem under consideration. The drawback of this method is that it does not work for mono-modal logics. In [tC05b], ten Cate uses a similar method to show that the elementarity problem for formulae in multi-modal languages with at least three different modalities is not even semi-decidable. It is not clear, however, if this can be generalized to mono-modal languages.

So, as corollary 1.5.2 shows, our plan for obtaining larger decidable approximations by closing under equivalence, is blocked in at least the two directions corresponding to frame and axiomatic equivalence.

In the rest of this section, we will provide a contrasting, positive answer as regards the decidability of classes obtained by closing under semantic equivalence. One caveat before we proceed — our purpose here is to show decidability, which, given the contrast with the undecidability result above, is already in itself of some significance. The efficiency of the algorithms we provide are not our primary concern here. (More efficient and elegant algorithms for deriving correspondence and canonicity are the subject matter of later chapters.) We will simply show that we can effectively look for an equivalent in the desired class, by providing an upper bound on the search space and then relying on the decidability of the equivalence in question and brute enumerations of the formulae in the space so bounded.

1.5.2 Semantic equivalence

In this section we will work in the basic modal language $\mathcal{L}$. Here we consider the closure of classes of formula (e.g. the Sahlqvist, Sahlqvist–van Benthem and monadic inductive formulae) under semantic equivalence. The strategy we will use to show that the obtained classes are decidable, will hinge on our being able to show that a formula is equivalent to some Sahlqvist formula if and only if it is equivalent to one with modal depth no greater than its own. We then argue that we can exhaustively check all candidates using the decidability of **K**.

Impervious formulae

Given a model $\mathcal{M} = (W, R, V)$, point $w \in W$, and a natural number $k \in \omega$, the *k-hull of $\mathcal{M}$ around w* is the submodel of $\mathcal{M}$ induced by the set of points in W which are reachable from w in no more that k R-steps. We will denote this by $\mathcal{M} \restriction_k w$.

Definition 1.5.5 Let $k \in \omega$. We say a formula φ is *k-impervious* if for all pointed models $(\mathcal{M}, m)$ it is the case that

$$(\mathcal{M}, m) \Vdash \varphi \text{ iff } ((\mathcal{M} \restriction_k m), m) \Vdash \varphi$$

Here are a few simple but useful observations regarding k-imperviousness:

- Obviously all formulas of depth at most k are k-impervious. The converse does not hold — the formula $\Box\Diamond\bot$ is 1-impervious.
- A formula is k-impervious if and only if it is l-impervious for all $l \geq k$.
- A formula φ is k-impervious if and only if $\neg\varphi$ is k-impervious.
- If φ and ψ are k-impervious formulae then so are $\varphi \wedge \psi$ and $\varphi \vee \psi$. The converse does not hold, take for example any instance of the tautology $p \vee \neg p$ or the contradiction $p \wedge \neg p$. These are all 0-impervious. Or consider the formula $p \wedge (p \vee \Diamond q)$ which is of depth 1 but 0-impervious.

- The truth of a k-impervious formula at a point in a model in not affected by any change (in the valuation, the domain or the accessibility relation) in the model beyond the k-hull.

Next we recall the definition of an n-bisimulation (see e.g. [BdRV01]).

Definition 1.5.6 Let $n \in \omega$, and $(\mathcal{M}, m) = ((W, R, V), m)$ and $(\mathcal{M}', m') = ((W', R', V'), m')$ be pointed models. We say $(\mathcal{M}, m)$ and $(\mathcal{M}', m')$ are *n-bisimilar* if there exists a sequence $Z_n \subseteq \cdots \subseteq Z_0 \subseteq W \times W'$ of binary relations between the domains of $(\mathcal{M}, m)$ and $(\mathcal{M}', m')$, satisfying, for $i < n$, the following

link mZ_nm',

local harmony if uZ_0v then u and v agree on the valuation of all proposition letters,

forth if $uZ_{i+1}u'$ and Ruv, then there exists a $v' \in W'$ such that $R'u'v'$ and vZ_iv',

back if $uZ_{i+1}u'$ and $Ru'v'$, then there exists a $v \in W$ such that Ruv and vZ_iv'.

We write $(\mathcal{M}, m) \rightleftarrows^n (\mathcal{M}', m')$ if $(\mathcal{M}, m)$ and $(\mathcal{M}', m')$ are n-bisimilar, or $(\mathcal{M}, m) \rightleftarrows^n_\Phi (\mathcal{M}', m')$ if the local harmony clause has been relativized with a set of propositional variables Φ.

We obtain the notion of *bisimulation* by replacing the sequence $Z_n \subseteq \cdots \subseteq Z_0$ in the above with a *single* relation $Z \subseteq W \times W'$. A bisimulation can be seen as the limiting case of an n-bisimulation. We write $(\mathcal{M}, m) \rightleftarrows (\mathcal{M}', m')$ if $(\mathcal{M}, m)$ and $(\mathcal{M}', m')$ are bisimilar.

The following proposition is standard:

Proposition 1.5.7 *Let Φ be a finite set of proposition letters. Then for all models $\mathcal{M}$ and $\mathcal{M}'$ and all points $m \in \mathcal{M}$ and $m' \in \mathcal{M}'$, the following are equivalent:*

1. $(\mathcal{M}, m) \rightleftarrows^n_\Phi (\mathcal{M}', m')$
2. *$(\mathcal{M}, m)$ and $(\mathcal{M}', m')$ agree on all formulas of $\mathcal{L}(\Phi)$ of modal depth at most n.*

Definition 1.5.8 The *depth of an occurrence of a subformula θ in a formula φ* is the number of occurrences of modalities in the scope of which θ occurs. Formally, we may define $depth(\theta, \varphi)$ by structural induction on φ as follows:

- $depth(\theta, \theta) = 0$;
- $depth(\theta, \neg\psi) = depth(\theta, \psi)$;
- $depth(\theta, \psi_1 \wedge \psi_2) = depth(\theta, \psi_1 \vee \psi_2) = depth(\theta, \psi_1)$ when the occurrence of θ under consideration is in ψ_1, and $depth(\theta, \psi_1 \wedge \psi_2) = depth(\theta, \psi_1 \vee \psi_2) = depth(\theta, \psi_2)$, otherwise;
- $depth(\theta, \Diamond\psi) = depth(\theta, \Box\psi) = 1 + depth(\theta, \psi)$.

Recall the definition of the *modal depth*, $depth(\varphi)$, of a formula φ (definition 1.3.1). Note that this is a special case of definition 1.5.8 — indeed, $depth(\varphi) = depth(\varphi, \varphi)$.

The operation on formulae which we now define will be used in reducing k-impervious formula to equivalent formula of corresponding modal depth.

Definition 1.5.9 For $\varphi \in \mathcal{L}$ and $k \in \omega$ define $\mathsf{CUT}_k(\varphi)$ to be the formula obtained from φ by

1. replacing every occurrence of a subformula of the form $\Diamond\psi$ such that $depth(\Diamond\psi, \varphi) = k$ with $\bot$, and
2. replacing every occurrence of a subformula of the form $\Box\psi$ such that $depth(\Box\psi, \varphi) = k$ with $\top$.

Note that $depth(\mathsf{CUT}_k(\varphi)) \leq k$. We will write $\mathsf{CUT}_k(\mathcal{L})$ for the set $\{\mathsf{CUT}_k(\varphi) \mid \varphi \in \mathcal{L}\}$. Equivalently, $\mathsf{CUT}_k(\mathcal{L})$ is the set of all $\mathcal{L}$-formulae with modal depth at most k.

The next definition and lemma are standard:

Definition 1.5.10 Let $\mathcal{M} = (W, R, V)$ be a model and $w \in W$. The *unravelling of* $\mathcal{M}$ *around* w is the model $\overrightarrow{\mathcal{M}} = (\overrightarrow{W}, \overrightarrow{R}, \overrightarrow{V})$ where

1. $\overrightarrow{W}$ is the set of all finite sequences $(v_0, v_1, \ldots, v_n)$ of elements of W such that $v_0 = w$ and $Rv_{i-1}v_i$ for all $1 \leq i \leq n$,
2. for all sequences $\overline{u}, \overline{v} \in \overrightarrow{W}$, it is the case that $\overrightarrow{R}\overline{u}\,\overline{v}$ if and only if $\overline{u}$ and $\overline{v}$ are of the form $(s_0, s_1, \ldots, s_n)$ and $(s_0, s_1, \ldots, s_n, s_{n+1})$ respectively, for some $n \in \omega$ (and, hence, Rs_ns_{n+1}),
3. for all sequences $\overline{u} \in \overrightarrow{W}$, it is the case that $\overline{u} \in \overrightarrow{V}(p)$ if and only if $\overline{u}$ is of the form $(s_0, s_1, \ldots, s_n)$ for some $n \in \omega$ with $s_n \in V(p)$.

The unravelling of a frame $\mathfrak{F} = (W, R)$ around a point $w \in W$ is obtained by simply dropping any reference to $\overrightarrow{V}$ in the above.

Lemma 1.5.11 *For any model $\mathcal{M} = (W, R, V)$ and $w \in W$ $(\mathcal{M}, w) \rightleftarrows (\overrightarrow{\mathcal{M}}, (w))$.*

Lemma 1.5.12 *Any k-impervious formula φ, is semantically equivalent to* $\mathsf{CUT}_k(\varphi)$.

Proof. Suppose $\varphi \in \mathcal{L}$ is k-impervious. Let $\mathcal{M}$ be any model and $w \in \mathcal{M}$. We show that $(\mathcal{M}, w) \Vdash \varphi$ if and only if $(\mathcal{M}, w) \Vdash \mathsf{CUT}_k(\varphi)$. Let $\overrightarrow{\mathcal{M}}$ be the unravelling of $\mathcal{M}$ around w. Then $(\mathcal{M}, w) \Vdash \varphi$ iff (by lemma 1.5.11) $(\overrightarrow{\mathcal{M}}, (w)) \Vdash \varphi$ iff (by the k-imperviousness of φ) $(\overrightarrow{\mathcal{M}} \upharpoonright_k (w), (w)) \Vdash \varphi$.

Now, when φ is evaluated at $(\overrightarrow{\mathcal{M}} \upharpoonright_k (w), (w))$, any occurrence of a subformula of φ at depth k in φ is evaluated at a point which is k steps from the root (w). But every such point is a dead-end (i.e. has no successors). Now, at dead-end points, any formula of the form $\Diamond\psi$ is false, while any formula of the form $\Box\psi$, is true. We conclude that $(\overrightarrow{\mathcal{M}} \upharpoonright_k (w), (w)) \Vdash \varphi$ iff and only if $(\overrightarrow{\mathcal{M}} \upharpoonright_k (w), (w)) \Vdash \mathsf{CUT}_k(\varphi)$. The conclusion follows. QED

Lemma 1.5.12 will be used to restrict the search space in terms of the depth of formulae considered as possible equivalents. The next lemma, which is needed to help make the search space finite, is an immediate consequence of the fact that semantic equivalence is preserved under uniform substitution of formulae for propositional variables.

Lemma 1.5.13 *Suppose $\varphi, \psi \in \mathcal{L}$ and $\varphi \equiv_{\text{sem}} \psi$. Then $\varphi \equiv_{\text{sem}} \psi(\bot/p)$ and $\varphi \equiv_{\text{sem}} \psi(\top/p)$, whenever $p \in \mathsf{PROP}(\psi) - \mathsf{PROP}(\varphi)$.*

Decidability of closure under semantic equivalence

We now apply lemmas 1.5.12 and 1.5.13 to show that it is decidable whether a given formula is a semantically equivalent to a Sahlqvist formula, or to an inductive formula.

Theorem 1.5.14 *Let $\mathfrak{C}$ be a class of $\mathcal{L}$-formulae such that:*

1. *if $\varphi \in \mathfrak{C}$ then $\varphi(\top/p) \in \mathfrak{C}$ or $\varphi(\bot/p) \in \mathfrak{C}$,*
2. *if $\varphi \in \mathfrak{C}$ then $\mathsf{CUT}_k(\varphi) \in \mathfrak{C}$ for all $k \in \omega$, and*
3. *given any finite set of proposition letters Φ and $k \in \omega$, we can effectively obtain a finite subset $C \subseteq \mathfrak{C}$, such that*
   (a) *$C \subseteq \mathsf{CUT}_k(\mathcal{L}(\Phi))$,*
   (b) *every formula $\varphi \in \mathfrak{C}$ with $\mathsf{PROP}(\varphi) \subseteq \Phi$ and $depth(\varphi) \leq k$, is semantically equivalent to a formula in C.*

Then the class of formulae $\mathfrak{C}^{sem}$, obtained by closing $\mathfrak{C}$ under semantic equivalence, is a decidable class of formulae.

Proof. Let $\mathfrak{C}$ be as in the formulation of the theorem. Let $\varphi \in \mathcal{L}$, and suppose $depth(\varphi) = k$ and $\mathsf{PROP}(\varphi) = \Phi$. Then we can determine whether or not $\varphi \in \mathfrak{C}^{sem}$ as follows: We first obtain a finite subset C of $\mathfrak{C}$ such that $C \subseteq \mathsf{CUT}_k(\mathcal{L}(\Phi))$, and every formula in $\mathfrak{C} \cap \mathsf{CUT}_k(\mathcal{L}(\Phi))$ is semantically equivalent to a formula in C.

By lemmas 1.5.12 and 1.5.13 and the assumption that $\mathfrak{C}$ is closed under the CUT_k-operation and under the substitution of either $\bot$ or $\top$ for propositional variables, it follows that if $\varphi \in \mathfrak{C}^{sem}$ if and only if φ is semantically equivalent to a formula in $\mathsf{CUT}_k(\mathcal{L}(\varphi)) \cap \mathfrak{C}$. Hence $\varphi \in \mathfrak{C}^{sem}$ if and only if φ is semantically equivalent to a formula in C.

Now given any two formulae $\psi_1, \psi_2 \in \mathcal{L}$, we can effectively decide whether $\psi_1 \equiv_{\text{sem}} \psi_2$ by checking whether $\mathbf{K} \vdash \psi_1 \leftrightarrow \psi_2$. The theorem follows. QED

Corollary 1.5.15 *The classes of formulae obtained by closing the classes of Sahlqvist, Sahlqvist–van Benthem and monadic inductive formulae under semantic equivalence, are decidable.*

Proof. The corollary follows from the fact that the classes of Sahlqvist, Sahlqvist–van Benthem and monadic inductive formulae satisfy the conditions imposed by theorem 1.5.14. Indeed, these classes are closed under the CUT_k-operation and under arbitrary substitutions of $\top$ and/or $\bot$ for propositional variables. Moreover, it is not difficult to set up a procedure which, given any finite set of propositional variables Φ and $k \in \omega$, constructs a finite set of Sahlqvist (respectively, Sahlqvist–van Benthem, respectively, monadic inductive) formulae containing, modulo semantic equivalence, every Sahlqvist (respectively, Sahlqvist–van Benthem, respectively, monadic inductive) formula in $\mathsf{CUT}_k(\mathcal{L}(\Phi))$.

We sketch one possible way of constructing this set in the case of the Sahlqvist formulae: First construct all positive $\mathcal{L}(\Phi)$-formulae of depth at most k, by forming all conjunctions of elements of Φ, and then taking all disjunctions of these. By bearing the associativity, commutativity or idempotency-laws in mind, there are only finitely many ($2^{2^{|\Phi|}}$, to be precise) formulae that we can construct in this way. Call the set so obtained S_0. Form the set S_1', consisting of all elements of S_0, all elements of S_0 prefixed with $\Diamond$, and all elements of S_0 prefixed with $\Box$. Now again form all conjunctions of elements of S_1' and all disjunctions of these, as before, yielding the set S_1, containing, up to semantic equivalence, all $\mathcal{L}(\Phi)$-formulae of depth at most 1. Note that $|S_1| = 2^{2^{|S_0|}}$. Continuing in this way up to stage k, we obtain, modulo semantic equivalence, all positive $\mathcal{L}(\Phi)$-formulae of depth at most k. The negative formulae and box formula can be similarly generated, as can the Sahlqvist antecedents, implications, and lastly the Sahlqvist formulae. Note that the sizes of the sets generated in this process form an exponential tower of height $2(k+1)+1$. QED

Chapter 2

The SQEMA-algorithm

In this chapter we introduce and study the SQEMA-algorithm, first introduced in [CGV06a]. SQEMA is an acronym for **S**econd-Order **Q**uantifier **E**limination for **M**odal Formulae using **A**ckermann's Lemma. This algorithm attempts the reduction of modal input formulae to equivalent $\mathcal{L}_r^n$, and hence L_0-formulae, through the application of a set of transformation rules. Most important amongst these transformation rules is the so-called Ackermann-rule, which allows for the elimination of propositional variables. This rule is based on a well-known lemma due to Ackermann. The other rules are based on simple modal and hybrid equivalences.

We illustrate the reach and limitations of the algorithm with several examples. It is shown that the algorithm is correct, and that all $\mathcal{L}$-formulae on which it succeeds are canonical. Next we investigate SQEMA's performance of well known syntactically specified classes of elementary and canonical formulae. Here it is shown that SQEMA successfully computes equivalents for all Sahlqvist, Sahlqvist–van Benthem and monadic inductive formulae. The elementarity and canonicity of the members of these classes then follows as a corollary.

SQEMA may be viewed as dealing primarily with the correspondence between $\mathcal{L}$ and $\mathcal{L}_r^n$, and only consequently with that between $\mathcal{L}$ and L_0. However, the equivalence between input formulae and the pure formulae which SQEMA obtains before translation into L_0, is rather subtle. In the concluding section we show how SQEMA can be extended and adapted to produce pure $\mathcal{L}_n^r$-formulae which are locally equivalent on frames to input formulae.

SQEMA will form the basis of our study of algorithmic correspondence and completeness in the chapters to come, where it will be extended and modified to produce correspondence and completeness results in richer languages as well as for ever larger classes of $\mathcal{L}$-formulae.

SQEMA has been implemented by Dimiter Georgiev ([Geo06]), and is available online.

2.1 Ackermann's lemma

The following is a well known lemma due to Ackermann ([Ack35]), here phrased for unary predicate variables P:

Lemma 2.1.1 (Ackermann's lemma) *Let P be a predicate variable and $A(\overline{z}, x)$ a first-order formula not containing P. Then, if $B(P)$ is negative in P, the equivalence*

$$\exists P \forall x((\neg A(\overline{z}, x) \vee P(x)) \wedge B(P)) \equiv B[A(\overline{z}, x)/P] \tag{2.1}$$

holds, with $B[A(\overline{z}, x)/P]$ the formula obtained by substituting $A(\overline{z}, x)$ for all occurrences P in B, the actual argument of each occurrence of P being substituted for x in $A(\overline{z}, x)$ every time. If $B(P)$ is positive in P, then the following equivalence holds:

$$\exists P \forall x((\neg P(x) \vee A(\overline{z}, x)) \wedge B(P)) \equiv B[A(\overline{z}, x)/P]. \tag{2.2}$$

The truth of the lemma rests on the idea of monotonicity:

Definition 2.1.2 A formula $\varphi \in \mathcal{L}_r^n$ is said to be *upward monotone* (respectively, *downward monotone*) in a propositional variable p, if $[\![\varphi]\!]_{\mathcal{M}} \subseteq [\![\varphi]\!]_{\mathcal{M}'}$ whenever $\mathcal{M} = (\mathfrak{F}, V)$ and $\mathcal{M}' = (\mathfrak{F}, V')$ are such that $V(p) \subseteq V'(p)$ (respectively, $V'(p) \subseteq V(p)$) and $V(q) = V'(q)$ for all propositional variables and nominals q other than p.

The following proposition, giving a syntactic sufficient condition for monotonicity, is easy to prove.

Proposition 2.1.3 *If a formula φ is positive (negative) in a propositional variable p, then is upward (downward) monotone in p.*

We can now formulate and prove a modal version of Ackermann's lemma:

Lemma 2.1.4 (Modal Ackermann lemma) *Let $A, B(p)$ be $\mathcal{L}_r^n$-formulae such that the propositional variable p does not occur in A and $B(p)$ is negative in p. Then for any model $\mathcal{M}$, it is the case that $\mathcal{M} \Vdash B(A)$ iff $\mathcal{M}' \Vdash (A \to p) \wedge B(p)$ for some model $\mathcal{M}'$ which may only differ from $\mathcal{M}$ on the valuation of p.*

Proof. If $\mathcal{M} \Vdash B(A)$, then $\mathcal{M}' \Vdash (A \to p) \wedge B(p)$ for a model $\mathcal{M}'$ such that $[\![p]\!]_{\mathcal{M}'} = [\![A]\!]_{\mathcal{M}}$. Conversely, if $\mathcal{M}' \Vdash (A \to p) \wedge B(p)$ for some model $\mathcal{M}'$ then $\mathcal{M}' \Vdash B(A/p)$ since $B(p)$ is downwards monotone in p. Therefore, $\mathcal{M} \Vdash B(A/p)$. QED

As in lemma 2.1.1, this lemma can be formulated for positive formulae B, too. This lemma is the core around which the SQEMA-algorithm is built.

We note that a somewhat different version of Ackermann's lemma has been proved for modal first-order formulae in [Sza02], where it is also applied to some modal formulae.

2.2 The Algorithm SQEMA

An expression of the form $\varphi \Rightarrow \psi$ with $\varphi, \psi \in \mathcal{L}_r^n$ is called a *SQEMA-sequent*. In a SQEMA-sequent $\varphi \Rightarrow \psi$, the formulae φ and ψ will be referred to as the *antecedent* and *consequent* of the sequent, respectively. A finite set of SQEMA-sequents is called a *SQEMA-system*. A sequent is *normalized* if both antecedent and consequent are in negation normal form. A system is *normalized* if every sequent in it is normalized.

Given a SQEMA-sequent $\varphi \Rightarrow \psi$, let $\mathsf{Form}(\varphi \Rightarrow \psi)$ be the formula $\varphi' \vee \psi'$, where φ' and ψ' are the formulae obtained by rewriting $\neg\varphi$ and ψ, respectively, into negation normal form. For a system of SQEMA-sequents $\mathsf{Sys} = \{\varphi_1 \Rightarrow \psi_1, \ldots, \varphi_n \Rightarrow \psi_n\}$, we define $\mathsf{Form}(\mathsf{Sys})$ to be the formula $\mathsf{Form}(\varphi_1 \Rightarrow \psi_1) \wedge \cdots \wedge \mathsf{Form}(\varphi_n \Rightarrow \psi_n)$. We will often write a system $\mathsf{Sys} = \{\varphi_1 \Rightarrow \psi_1, \ldots, \varphi_n \Rightarrow \psi_n\}$ in the form

$$\left\|\begin{array}{l}\varphi_1 \Rightarrow \psi_1 \\ \vdots \\ \varphi_n \Rightarrow \psi_n\end{array}\right. .$$

A system Sys is *positive* (negative) in a propositional variable p if Form(Sys) is positive (negative) in p. A system Sys is called *pure* if Form(Sys) is a pure hybrid formula. In general, whenever we attribute a property, usually predicated of formulae, to a system Sys, we will mean that Form(Sys) has that property.[1]

A sequent of the form $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$ with $\mathbf{j}$ and $\mathbf{k}$ nominals, will be called a *diamond-link sequent*. All other sequents will be referred to as *non-diamond-link sequents*.

2.2.1 The transformation rules of SQEMA

The transformation rules used by the algorithm SQEMA are listed below. Concerning these rules, we note the following:

1. Among the rules for propositional connectives, the only rule that could yield a sequent which is not normalized, when applied to a normalized sequent, is the $\vee$-rule.

2. Apart from the polarity switching-rule, no transformation rule changes the polarity of any occurrence of a propositional variable within a system.

Rules for the logical connectives

These are the rules used to bring a system into the form to which the Ackermann-rule (see below) is applicable. Note that these rules are applied to individual sequents within systems. Moreover they are *rewriting rules*, i.e., the sequent above the line is replaced in the system by the sequent(s) listed below the line.

[1] SQEMA-sequents are called SQEMA-equations in [CGV06a]. This name was used because of a ceratin resemblance between the execution of SQEMA on formula and the procedure of solving systems of linear equations by Gaussian elimination.

$$\frac{C \Rightarrow (A \wedge B)}{C \Rightarrow A, C \Rightarrow B} \tag{$\wedge$-rule}$$

$$\frac{C \Rightarrow (A \vee B)}{(C \wedge \neg A) \Rightarrow B} \tag{Left-shift $\vee$-rule}$$

$$\frac{(C \wedge A) \Rightarrow B}{C \Rightarrow (\neg A \vee B)} \tag{Right-shift $\vee$-rule}$$

$$\frac{A \Rightarrow \Box B}{\Diamond^{-1} A \Rightarrow B} \tag{$\Box$-rule}$$

$$\frac{\Diamond^{-1} A \Rightarrow B}{A \Rightarrow \Box B} \tag{Inverse $\Diamond$-rule}$$

Lastly, the diamond-rule,

$$\frac{\mathbf{j} \Rightarrow \Diamond A}{\mathbf{j} \Rightarrow \Diamond \mathbf{k},\ \mathbf{k} \Rightarrow A} \tag{$\Diamond$-rule}$$

where the antecedent $\mathbf{j}$ is a nominal, and $\mathbf{k}$ is a new nominal not occurring in the system.

Ackermann-rule

This rule is based on the equivalence given in Ackermann's lemma. It works not on a single equation, but an entire system, as follows:

$$\text{The system} \quad \left\| \begin{array}{l} A_1 \Rightarrow p, \\ \vdots \\ A_n \Rightarrow p, \\ B_1(p), \\ \vdots \\ B_m(p), \end{array} \right. \quad \text{is replaced by} \quad \left\| \begin{array}{l} B_1((A_1 \vee \ldots \vee A_n)/p), \\ \vdots \\ B_m((A_1 \vee \ldots \vee A_n)/p). \end{array} \right.$$

where:

1. p does not occur in $A_1, \ldots, A_n$;
2. Each of $B_1, \ldots, B_m$ is negative in p, or does not contain p at all.

Polarity switching rule

This rule is applied to a normalized system as a whole: Substitute $\neg p$ for every occurrence of p in the system.

Normalization rules

These rules are applied only within antecedents and consequents of sequents, and are used, by their exhaustive application, to bring antecedents and consequents into negation normal form:

1. Replace $\neg\Diamond\varphi$ with $\Box\neg\varphi$, and $\neg\Diamond^{-1}\varphi$ with $\Box^{-1}\neg\varphi$;
2. Replace $\neg\Box\varphi$ with $\Diamond\neg\varphi$, and $\neg\Box^{-1}\varphi$ with $\Diamond^{-1}\neg\varphi$;
3. Replace $\neg(\varphi \wedge \psi)$ with $(\neg\varphi \vee \neg\psi)$;
4. Replace $\neg(\varphi \vee \psi)$ with $(\neg\varphi \wedge \neg\psi)$.

Auxiliary rules

These rules are intended to provide the algorithm with some (extremely basic) propositional reasoning capabilities. Like normalization rules, they are applied within antecedents and consequents of sequents. Of course more rules can be added, or one could even introduce a complete deductive system for propositional logic. See for example the richer set of rules in chapter 16 of [GSS06]. For the illustrative purposes we have in mind however, these three rules will suffice.

1. Replace $\gamma \vee \neg\gamma$ with $\top$, and $\gamma \wedge \neg\gamma$ with $\bot$.
2. Replace $\gamma \vee \top$ with $\top$, and $\gamma \vee \bot$ with γ.
3. Replace $\gamma \wedge \top$ with γ, and $\gamma \wedge \bot$ with $\bot$.

2.2.2 Specification of the algorithm

Here follows a pseudo code specification of the SQEMA-algorithm. We provide slightly more detail than is strictly necessary for our purposes, which include the proving of correctness (section 2.4), completeness (section 2.6) and canonicity (section 2.5) results. Even so, this level of detail is still probably quite insufficient for implementation purposes, with which we will not be concerned. A modified version of SQEMA, called SQEMA$^+$, providing more structure (e.g. with built in preference ordering on the application of transformation rules) is introduced in chapter 16 of [GSS06].

Algorithm SQEMA(φ). This is the main body of the algorithm. It takes an $\mathcal{L}_r^n$-formula as an input and either returns a first-order local equivalent for the input formula, or reports failure.

Phase 1: Preprocessing. Call subroutine Preprocess(φ), to be introduced below. It returns a modal formula $\bigvee \alpha_k$ semantically equivalent to $\neg\varphi$.

Phase 2: Elimination of Propositional Variables.

2.1 For each disjunct α_k of the formula $\bigvee \alpha_k$ returned by $\mathsf{Preprocess}$, form the *initial system* $\|\mathbf{i} \Rightarrow \alpha_k$, where $\mathbf{i}$ is a fixed, reserved nominal, not allowed to occur in any input formula. Then call subroutine $\mathsf{Transform}(\|\mathbf{i} \Rightarrow \alpha_k)$.

2.2 If $\mathsf{Transform}(\|\mathbf{i} \Rightarrow \alpha_k)$ returns FAIL for any α_k, return FAIL and terminate, else, proceed to phase 3.

Phase 3: Postprocessing and Translation. If this phase it reached it means that, for every k, the subroutine $\mathsf{Transform}(\|\mathbf{i} \Rightarrow \alpha_k)$ has succeeded and has returned a pure system Sys_k. Continue as follows:

3.1: Form the set $\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$ of all pure systems returned by the subroutine $\mathsf{Transform}(\|\mathbf{i} \Rightarrow \alpha_k)$.

3.2: Call $\mathsf{Postprocess}(\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\})$.

3.3: The subroutine $\mathsf{Postprocess}(\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\})$ produces a first-order formula. Return this formula and terminate.

Preprocess(φ). This subroutine preprocesses the formula φ by negating it, transforming it into negation normal form, and 'bubbling up' the disjunctions.

Preprocess.1: Negation and Normal Form. Negate φ and rewrite $\neg\varphi$ in negation normal form by eliminating the connectives '$\rightarrow$' and '$\leftrightarrow$', and by driving all negation signs inwards until they appear only directly in front of propositional variables and/or nominals.

Preprocess.2: Bubbling up Disjunctions. Distribute diamonds and conjunctions over disjunctions as much as possible, using the equivalences $\Diamond(\psi \vee \theta) \equiv \Diamond\psi \vee \Diamond\theta$ and $(\varphi \vee \psi) \wedge \theta \equiv (\varphi \wedge \theta) \vee (\psi \wedge \theta)$, in order to obtain a formula of the form $\bigvee \alpha_k$, where no further distribution of diamonds and conjunctions over disjunctions is possible in any α_k.

Transform(Sys) The aim of this procedure is to eliminate all occurring propositional variables from the input system Sys, if possible, and to return a pure system.

Transform.1: Eliminate every propositional variable in which the system is positive or negative, by substituting it with $\top$ or $\bot$, respectively.

Transform.2: While the system Sys is not pure (i.e., it contains equations that contain propositional variables), choose a propositional variable, say p, to eliminate, and call $\mathsf{Eliminate}(\mathsf{Sys}, p)$.

Transform.3: If $\mathsf{Eliminate}(\mathsf{Sys}, p)$ has returned FAIL for every variable p remaining in the Sys, return FAIL

else, if $\mathsf{Eliminate}(\mathsf{Sys}, p)$ returns a system Sys' (in which p has been eliminated),

Transform.3.1: Call $\mathsf{Transform}(\mathsf{Sys}')$

Transform.3.2: if $\mathsf{Transform}(\mathsf{Sys}')$ returns FAIL, return FAIL
else, if $\mathsf{Transform}(\mathsf{Sys}')$ returns a pure system, Sys'', return Sys''.

Eliminate(Sys, p) This procedure takes as input a SQEMA-system Sys together with a propositional variable. The goal is, by applying the *transformation rules for the propositional connectives* (subsection 2.2.1), to rewrite the system so that the Ackermann-rule becomes applicable with respect to the chosen variable p in order to eliminate it. Thus, the current goal is to transform the system into one in which every sequent is either negative in p, or of the form $\alpha \Rightarrow p$, with p not occurring in α, i.e., to 'extract' p or 'solve' for it. If this can be achieved, the Ackermann-rule is applied, eliminating the variable p.

After the application of the Ackermann-rule, the system is re-normalized by applying the *normalization rules* (subsection 2.2.1). The application of these normalization rules is interleaved with that of the *auxiliary rules* (subsection 2.2.1). Specifically, after each application of a normalization rule, the auxiliary rules are applied to simplify the system, if possible.

If the this process succeeds in eliminating the designated propositional variable, Eliminate returns the transformed system Sys' from which p has been eliminated; else, Eliminate returns FAIL.

Postprocessing($\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$) This procedure receives a set of pure systems from which it computes and returns a first-order formula.

Postprocessing.1: For each $\mathsf{Sys}_k \in \{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$, let pure_k be the pure formula $\mathsf{Form}(\mathsf{Sys}_k)$.

Postprocessing.2: Form the formula $\mathsf{pure}(\varphi)$ by taking the disjunction of the formulae pure_k, obtained in step Postprocessing.1

Postprocessing.3: Form the formula $\forall \overline{y} \exists x \mathrm{ST}(\neg \mathsf{pure}(\varphi), x)$, where $\overline{y}$ is the tuple of all occurring variables corresponding to nominals, but with $y_{\mathbf{i}}$ (corresponding to the designated current state nominal $\mathbf{i}$) left free, since a local correspondent is being computed. Return this first-order formula.

Remark 2.2.1 Some comments are in order:

1. Propositional variables are eliminated from systems of SQEMA-equations one at a time. The choice of the next variable to be eliminated (in Transform.2) depends on the strategy being followed. We will not discuss such ordering-strategies here, but assume that the choice is made nondeterministically. Notice that the way in which subroutine Transform calls itself recursively ensures that all possible orders of elimination are explored until either an order that succeeds is found, or all orders have failed.

2. Although we concentrate on input formulae from the basic modal langauge $\mathcal{L}$ in this chapter, the algorithm, as described, accepts any $\mathcal{L}_r^n$ formula as input.

2.3 Examples

In this section we illustrate the algorithm by giving examples of its execution on various input formulae, and discus some of its features. In order to enhance the readability, we will not be over scrupulous in showing each and every step of the algorithm, as given in section 2.2.2.

Example 2.3.1 Consider the Geach formula, $\Diamond\Box p \rightarrow \Box\Diamond p$.

Phase 1 Preprocessing yields the formula: $\Diamond\Box p \wedge \Diamond\Box\neg p$.

Phase 2 The execution does not branch due to disjunctions, hence there is only one initial system, namely

$$\|\mathbf{i} \Rightarrow (\Diamond\Box p \wedge \Diamond\Box\neg p)\ .$$

The system is neither positive nor negative in p, the only occurring propositional variable. We choose p to eliminate — our only option. We will try to transform the system, using the rules, so that the Ackermann-rule becomes applicable. Applying the $\wedge$-rule gives

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\Box p \\ \mathbf{i} \Rightarrow \Diamond\Box\neg p \end{array} \right. ,$$

Applying first the $\Diamond$-rule and then the $\Box$-rule to the first sequent yields:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{j} \Rightarrow \Box p \\ \mathbf{i} \Rightarrow \Diamond\Box\neg p \end{array} \right. ,$$

and then

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \Diamond^{-1}\mathbf{j} \Rightarrow p \\ \mathbf{i} \Rightarrow \Diamond\Box\neg p \end{array} \right. .$$

The Ackermann-rule is now applicable, yielding the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{i} \Rightarrow \Diamond\Box\neg(\Diamond^{-1}\mathbf{j}) \end{array} \right. ,$$

which the application of the normalization (and auxiliary) rules simplifies to

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{i} \Rightarrow \Diamond\Box\Box^{-1}\neg\mathbf{j} \end{array} \right. .$$

As all propositional variables have now been eliminated, phase 2 terminates successfully, and the algorithm proceeds to phase 3.

Phase 3 With Sys the above pure system, $\mathsf{Form}(\mathsf{Sys})$ is the formula

$$(\neg\mathbf{i} \vee \Diamond\mathbf{j}) \wedge (\neg\mathbf{i} \vee \Diamond\Box\Box^{-1}\neg\mathbf{j}).$$

Negating (and applying the definition of $\rightarrow$ to enhance readability) we obtain

$$(\mathbf{i} \rightarrow \Diamond\mathbf{j}) \rightarrow (\mathbf{i} \wedge \Box\Diamond\Diamond^{-1}\mathbf{j}),$$

which, translated, becomes

$$\forall y_{\mathbf{j}} \exists x_0 [R y_{\mathbf{i}} y_{\mathbf{j}} \to (x_0 = y_{\mathbf{i}}) \wedge \forall y (R x_0 y \to \exists u (R y u \wedge \exists v (R v u \wedge v = y_{\mathbf{j}})))],$$

and simplifies to

$$\forall y_{\mathbf{j}} [R y_{\mathbf{i}} y_{\mathbf{j}} \to \forall y (R y_{\mathbf{i}} y \to \exists u (R y u \wedge R y_{\mathbf{j}} u))],$$

defining the Church-Rosser property, as expected. Note that the variable $y_{\mathbf{i}}$ occurs free and corresponds to the nominal $\mathbf{i}$, which we interpret as the current state. Hence we obtain a *local* property. Also, we directly translate $\mathbf{i} \to \Diamond \mathbf{j}$ as $R y_{\mathbf{i}} y_{\mathbf{j}}$, since $\mathrm{ST}(\mathbf{i} \to \Diamond \mathbf{j}, x_0)$ is $x_0 = y_{\mathbf{i}} \to \exists z (R x_0 z \wedge z = y_{\mathbf{j}})$, which clearly simplifies.

$\lhd$

Example 2.3.2 Consider the formula $p \wedge \Box(\Diamond p \to \Box q) \to \Diamond \Box \Box q$, from example 1.3.20. Recall that this formula is not equivalent to any Sahlqvist formula.

Phase 1 Preprocessing yields $p \wedge \Box(\Box \neg p \vee \Box q) \wedge \Box \Diamond \Diamond \neg q$.

Phase 2 We have one initial system, namely $\| \mathbf{i} \Rightarrow [p \wedge \Box(\Box \neg p \vee \Box q) \wedge \Box \Diamond \Diamond \neg q]$. This system is neither positive nor negative in p or q. Choose p to eliminate. Applying the $\wedge$-rule twice, we get

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow p \\ \mathbf{i} \Rightarrow \Box(\Box \neg p \vee \Box q) \\ \mathbf{i} \Rightarrow \Box \Diamond \Diamond \neg q \end{array} \right. .$$

The system is now ready for the application of the Ackermann-rule, as p has been successfully isolated, and $\mathbf{i} \Rightarrow \Box(\Box \neg p \vee \Box q)$ and $\mathbf{i} \Rightarrow \Box \Diamond \Diamond \neg q$ are negative in p. This yields

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box \neg \mathbf{i} \vee \Box q) \\ \mathbf{i} \Rightarrow \Box \Diamond \Diamond \neg q \end{array} \right. .$$

No normalization or auxiliary rules are applicable. The only remaining variable to be eliminated is q. Successively applying the $\Box$-rule, left-shift $\vee$-rule and $\Box$-rule again, we get

$$\left\| \begin{array}{l} \Diamond^{-1}(\Diamond^{-1} \mathbf{i} \wedge \neg \Box \neg \mathbf{i}) \Rightarrow q \\ \mathbf{i} \Rightarrow \Box \Diamond \Diamond \neg q \end{array} \right. .$$

Applying the Ackermann-rule to eliminate q, yields:

$$\| \ \mathbf{i} \Rightarrow \Box \Diamond \Diamond \neg [\Diamond^{-1}(\Diamond^{-1} \mathbf{i} \wedge \neg \Box \neg \mathbf{i})] \ ,$$

which the application of the normalization rules turns into

$$\| \ \mathbf{i} \Rightarrow \Box \Diamond \Diamond [\Box^{-1}(\Box^{-1} \neg \mathbf{i} \vee \Box \neg \mathbf{i})] \ .$$

As all propositional variables have been eliminated, we proceed to phase 3.

Phase 3 We see that, with φ the input formula under consideration in this example, $(\neg\mathbf{i} \vee \Box\Diamond\Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \Box\neg\mathbf{i}))$ is the formulae $\mathsf{pure}(\varphi)$, obtained in Postprocessing.2. Negated this becomes $\mathbf{i} \wedge \Diamond\Box\Box[\Diamond^{-1}(\Diamond^{-1}\mathbf{i} \wedge \Diamond\mathbf{i})$. Translating into first-order logic we obtain

$$\begin{aligned}&\exists x_0[x_0 = y_{\mathbf{i}} \wedge \exists z_1(Rx_0z_1 \wedge \forall z_2(Rz_1z_2 \to \forall z_3(Rz_2z_3 \to \exists u_1[Ru_1z_3\\ &\wedge \exists u_2(Ru_2u_1 \wedge u_2 = y_{\mathbf{i}}) \wedge \exists u_3(Ru_1u_3 \wedge u_3 = y_{\mathbf{i}})]))))].\end{aligned}$$

Note that, in this example, the order of elimination of the propositional variables is inessential, as eliminating q first and then p works equally well. $\lhd$

Example 2.3.3 Consider the formula $\Box(\Box p \leftrightarrow q) \to p$. The current implementations of both SCAN and DLS fail on this formula. Let's see what SQEMA does with it:

Phase 1 Preprocessing yields $\Box((\Diamond\neg p \vee q) \wedge (\neg q \vee \Box p)) \wedge \neg p$.

Phase 2 Again we have only one initial system:

$$\big\| \ \mathbf{i} \Rightarrow \Box((\Diamond\neg p \vee q) \wedge (\neg q \vee \Box p)) \wedge \neg p \quad .$$

All variables occur both positively and negatively. Choose q to eliminate. Applying the $\wedge$-rule and the $\Box$-rule we transform the system into

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\Diamond\neg p \vee q) \\ \Diamond^{-1}\mathbf{i} \Rightarrow (\neg q \vee \Box p) \\ \mathbf{i} \Rightarrow \neg p \end{array} \right. .$$

Applying the left-shift $\vee$-rule to the first sequent yields

$$\left\| \begin{array}{l} (\Diamond^{-1}\mathbf{i} \wedge \neg\Diamond\neg p) \Rightarrow q \\ \Diamond^{-1}\mathbf{i} \Rightarrow (\neg q \vee \Box p) \\ \mathbf{i} \Rightarrow \neg p \end{array} \right. ,$$

to which the Ackermann-rule is applicable with respect to q. This gives

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\neg(\Diamond^{-1}\mathbf{i} \wedge \neg\Diamond\neg p) \vee \Box p) \\ \mathbf{i} \Rightarrow \neg p \end{array} \right. .$$

Now, during the application of the normalization rules to the system, working outwards from subformulae to superformulae, the system

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\neg\Diamond^{-1}\mathbf{i} \vee \neg\Box p \vee \Box p) \\ \mathbf{i} \Rightarrow \neg p \end{array} \right.$$

is obtained. Recall that the application of auxiliary rules is interleaved with that of the normalization rules. In this case the auxiliary rules $\gamma \vee \neg\gamma \equiv \top$ and $\gamma \vee \top \equiv \top$ now transform the above system into

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow \top \\ \mathbf{i} \Rightarrow \neg p \end{array} \right.$$

The only remaining propositional variable to be eliminated is p. But the system is negative in p. Hence $\bot$ is substituted for all occurrences of p, yielding the system

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow \top \\ \mathbf{i} \Rightarrow \neg\bot \end{array} \right. ,$$

which is pure.

Phase 3 We obtain $(\neg\Diamond^{-1}\mathbf{i} \vee \top) \wedge (\neg\mathbf{i} \vee \neg\bot)$ for $\mathsf{pure}(\varphi)$, which becomes $(\Diamond^{-1}\mathbf{i} \wedge \bot) \vee (\mathbf{i} \wedge \bot)$ after negation. After translation and simplification we obtain the first-order equivalent $\bot$.

Some remarks are in order here. Firstly, note that the success of the algorithm may depend essentially upon the ability to do some propositional reasoning, as supplied by the auxiliary rules in conjunction with the normalization rules. Particularly, from the sequent $\Diamond^{-1}\mathbf{i} \Rightarrow (\neg(\Diamond^{-1}\mathbf{i} \wedge \neg\Diamond\neg p) \vee \Box p)$ we had to obtain $\Diamond^{-1}\mathbf{i} \Rightarrow (\neg\Diamond^{-1}\mathbf{i} \vee \neg\Box p \vee \Box p)$. This was possible by applying the normalization rules from inside outwards. It is easy to check that, had these rules been applied inwards, i.e. working from superformulae to subformulae, this form would not have been obtained, and the tautological status of the consequent of the sequent would not have been captured by the auxiliary rules. Particularly, the obtained system would have been

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\Box^{-1}\neg\mathbf{i} \vee \Diamond\neg p \vee \Box p) \\ \mathbf{i} \Rightarrow \neg p \end{array} \right. .$$

It is clear that the further application of SQEMA-transformation rules will fail transform system into one to which the Ackermann-rule is applicable.

A much more robust approach would be to replace the Ackermann-rule with a stronger version, which involves testing for *monotonicity*, rather than polarity (this option is explored in chapter 6). Indeed, testing the consequent of the sequent for (upward) monotonicity would give the answer 'yes' — this is, post hoc, easy to see, since we already know that the consequent is a tautology. This would allow us, after changing the polarity of p, to apply the stronger, monotonicity based, version of the Ackermann-rule to the system

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\Box^{-1}\neg\mathbf{i} \vee \Diamond p \vee \Box\neg p) \\ \mathbf{i} \Rightarrow p \end{array} \right. ,$$

and hence to obtain

$$\left\| \; \Diamond^{-1}\mathbf{i} \Rightarrow (\Box^{-1}\neg\mathbf{i} \vee \Diamond\mathbf{i} \vee \Box\neg\mathbf{i}) \right. ,$$

which, after negation and translation, is equivalent to $\bot$, as before.

Now a second point to take note of. Suppose that we had tried to eliminate p first. Note that we will gain nothing by changing the polarity of p for we cannot get the occurrence of p 'out' under the diamond in the sequent $\Diamond^{-1}\mathbf{i} \Rightarrow (\Diamond p \vee q)$. Indeed, the system may be transformed to become

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\Diamond\neg p \vee q) \\ \Diamond^{-1}(\Diamond^{-1}\mathbf{i} \wedge \neg\neg q) \Rightarrow p \\ \mathbf{i} \Rightarrow \neg p \end{array} \right. ,$$

to which we may apply the Ackermann-rule with respect to p and obtain

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow (\Diamond\neg\Diamond^{-1}(\Diamond^{-1}\mathbf{i} \wedge \neg\neg q) \vee q) \\ \mathbf{i} \Rightarrow \neg\Diamond^{-1}(\Diamond^{-1}\mathbf{i} \wedge \neg\neg q) \end{array} \right. .$$

It is clear that this system cannot be solved for q by the application of SQEMA-transformation rules — the positive and negative occurrences of q separated by a disjunction in the consequent of the first sequent make this impossible. Moral: the order of elimination does matter sometimes, that is why the algorithm incorporates the ability to backtrack and explore different, and eventually all, orders of elimination. $\lhd$

Example 2.3.4 Consider the conjunction of the formula $(\Box p \to \Box\Box p) \wedge \Box(\Box p \to \Box\Box p)$ with the McKinsey formula $\Box\Diamond p \to \Diamond\Box p$. Recall that, according to proposition 1.3.27, this formula is not a van Benthem-formula. This formula defines a first-order property on frames ([vB76]) and also happens to be canonical but, as we will now see, SQEMA fails is.

Phase 1 Preprocessing yields $(\Box p \wedge \Diamond\Diamond\neg p) \vee \Diamond(\Box p \wedge \Diamond\Diamond\neg p) \vee (\Box\Diamond p \wedge \Box\Diamond\neg p)$.

Phase 2 We obtain three initial systems, namely $\|\mathbf{i} \Rightarrow (\Box p \wedge \Diamond\Diamond\neg p)$, $\|\mathbf{i} \Rightarrow \Diamond(\Box p \wedge \Diamond\Diamond\neg p)$ and $\|\mathbf{i} \Rightarrow (\Box\Diamond p \wedge \Box\Diamond\neg p)$. It is easy to see that p will be successfully eliminated from the first two systems. In the third system, the only applicable rules are the $\wedge$ and $\Box$-rules, the application of which yields a system

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \Rightarrow \Diamond p \\ \Diamond^{-1}\mathbf{i} \Rightarrow \Diamond\neg p \end{array} \right. .$$

The only rules that are applicable to this system are the inverse $\Diamond$-rule and the polarity switching rule, and no combination of these will transform the system into one to which the Ackermann-rule is applicable. The algorithm therefore reports failure.

Note that the system on which the algorithm fails is identical to the system on which it would fail if we had given it as input the McKinsey formula alone. Indeed, we should expect failure in that case, since the McKinsey formula does not have a first-order frame correspondent. The point we wish to emphasize is that the separate processing of the disjuncts of the preprocessed formula seems to preclude SQEMA's success in this, and similar, cases. $\lhd$

Example 2.3.5 The logic axiomatized with the formula $(\Box p \to \Box\Box p) \wedge (\Box\Diamond p \to \Diamond\Box p)$ from example 2.3.4, also known as $\mathbf{K4.1} = \mathbf{K4} \oplus (\Box\Diamond p \to \Diamond\Box p)$, is an example of what is known as a *co-final subframe logic*. That is to say the class of Kripke frames for this logic is closed under taking co-final subframes (a certain type of not necessarily generated subframes). If the class of frames for a logic is closed under taking arbitrary subframes, then it is called a *subframe logic*. Thus the subframe logics form a subclass of the co-final subframe logics.

Many well known logics are in fact co-final subframe logics, and those above **K4** turn out to be particularly well behaved (see [Fin85], [Zak96], and [Zak97b]). Specifically, for these logics the properties of elementarity, canonicity, strong Kripke completeness and compactness coincide.

Some examples are $\mathbf{S4} = \mathbf{K4} \oplus p \to \Diamond p$, $\mathbf{GL} = \mathbf{K4} \oplus \Box(\Box p \to p) \to \Box p$, $\mathbf{S4.2} = \mathbf{S4} \oplus \Diamond\Box p \to \Box\Diamond p$. For many more examples see [Zak97b]. Now the axioms of $\mathbf{GL}$ are non-elementary, and one can easily check that SQEMA will fail on their conjunction, as indeed it should. SQEMA also fails on the conjunction of the axioms of $\mathbf{K4.1}$, as seen above in example 2.3.4, which, as mentioned, is in fact elementary and canonical.

On the other hand it is easy to check that SQEMA will succeed on the conjunctions of the axioms of $\mathbf{S4}$ and $\mathbf{S4.2}$, respectively, as well as on those of the majority the well-know (co-final) subframe logics as listed e.g. in [Zak97b].

It has been shown in [Zak96] (see also [Zak97b]) that every (co-final) subframe logic above $\mathbf{K4}$ is axiomatizable with certain types of *canonical formulas* in the sense of [Zak92] (not to be confused with formulae which are canonical in the sense in which we employ the term), known as *(co-final) subframe formulas*. The subrame formulae have the form

$$\alpha(\mathfrak{F}) = \bigwedge_{a_i R a_j} \alpha_{ij} \wedge \bigwedge_{i=0}^{n} \alpha_i \to p_0,$$

while the co-final subrame formulae have the form

$$\alpha(\mathfrak{F}, \bot) = \bigwedge_{a_i R a_j} \alpha_{ij} \wedge \bigwedge_{i=0}^{n} \alpha_i \wedge \alpha_\bot \to p_0,$$

where

$$\alpha_{ij} = \Box^+(\Box p_j \to p_i),$$

$$\alpha_i = \Box^+((\bigwedge_{\neg a_i R a_k} \Box p_k \wedge \bigwedge_{j \neq i} p_j \to p_i) \to p_i),$$

$$\alpha_\bot = \Box^+(\bigwedge_{i=0}^{n} \Box^+ p_i \to \bot),$$

$\mathfrak{F} = (W, R)$ is a finite frame with $W = \{a_0, \ldots, a_n\}$, and $\Box^+\varphi$ is shorthand for $\varphi \wedge \Box\varphi$.

Now every subframe logic L (respectively, co-final subframe logic L) above $\mathbf{K4}$ can be given as $L = \mathbf{K4} \oplus \{\alpha(\mathfrak{F}_i) \mid i \in I\}$ (respectively, $L = \mathbf{K4} \oplus \{\alpha(\mathfrak{F}_i, \bot) \mid i \in I\}$), for some family of finite frames $\{\mathfrak{F}_i \mid i \in I\}$. For example $\mathbf{S4} = \mathbf{K4} \oplus \alpha(\mathfrak{F})$ where $\mathfrak{F}$ is the frame consisting of a single, irreflexive point. In other words

$$\mathbf{S4} = \mathbf{K4} \oplus \Box^+((\Box p_0 \to p_0) \to p_0) \to p_0.$$

Let us see what SQEMA will do with the conjunction of these axioms. Negating and rewriting in negation normal form we obtain

$$[\Box p \wedge \Diamond\Diamond\neg p] \vee [((\Box p_0 \wedge \neg p_0) \vee p_0) \wedge \Box((\Box p_0 \wedge \neg p_0) \vee p_0) \wedge \neg p_0],$$

which becomes

$$[\Box p \wedge \Diamond\Diamond\neg p] \vee [(\Box p_0 \wedge \neg p_0) \wedge \Box((\Box p_0 \wedge \neg p_0) \vee p_0) \wedge \neg p_0] \vee [p_0 \wedge \Box((\Box p_0 \wedge \neg p_0) \vee p_0) \wedge \neg p_0]$$

after bubbling up disjunctions. The execution will thus proceed along three disjunctive branches. Let us consider the branch corresponding to the second disjunct above. After the application of the $\wedge$ and $\Box$-rules the system on that branch has the form

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box p_0 \wedge \neg p_0 \\ \Diamond^{-1}\mathbf{i} \Rightarrow (\Box p_0 \wedge \neg p_0) \vee p_0 \\ \mathbf{i} \Rightarrow \neg p_0 \end{array} \right. .$$

It should be clear that the algorithm will fail, since the negative and positive occurrences of p_0 in the second sequent cannot be separated by the application of transformation rules. This type of failure can sometimes be avoided be distributing disjunctions in the scope of boxes over conjunctions, during the preprocessing stage. But in the present case this would yield a system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box p_0 \wedge \neg p_0 \\ \mathbf{i} \Rightarrow \Box((\Box p_0 \vee p_0) \wedge (\neg p_0 \vee p_0)) \\ \mathbf{i} \Rightarrow \neg p_0 \end{array} \right.$$

on which the algorithm would also fail.

Inspecting the definition of the formulae $\alpha(\mathfrak{F})$ and $\alpha(\mathfrak{F}, \perp)$ a bit closer one sees that SQEMA will *always* fail on them. Indeed if $\mathfrak{F}$ contains a reflexive point a_i then the negation of $\alpha(\mathfrak{F})$ (and of $\alpha(\mathfrak{F}, \perp)$) will contain a subformula $\Box(\Diamond\neg p_i \vee p_i)$ (origination from α_{ii}) and if it contains an irreflexive point we will be in a situation similar to the one illustrated above. This is rather a pity considering that the (co-final) subframe logics have so many desirable properties and also that SQEMA would succeed on the standard axiomatizations of many well known (co-final) subframe logics. It would be interesting to see if either (i) the (co-final) subframe formulae could be redefined in an equivalent way which would allow SQEMA to succeed on them more often, or (ii) alternatively an algorithm could be developed to compute first-order equivalents for elementary (co-final) subframe formulae. $\lhd$

Remark 2.3.6 We conclude this section with some informal remarks about the computational complexity of the SQEMA. The algorithm as it stands is rather loosely specified — no mechanism for determining which rules to apply when or in what order to try and eliminate propositional variables is specified — hence we will content ourselves with sketching a rough upper bound.

Let us consider the execution of the algorithm on an input formula φ. Suppose that φ contains n symbols. During preprocessing this formula is negated and disjunctions are 'bubbled up'. The latter can in general lead to an at most single exponential blowup in the size of the formula, and can be completed in no more than 2^n steps.

Each disjunct of the preprocessed input formula gives rise to a system from which the propositional variables are to be eliminated. We may assume that every transformation rule takes constant time to apply. Moreover, since each transformation rule corresponds to the occurrence in the system of a connective, the number of transformation rule-applications during the process of solving for a specific variable p is bounded by the number of connectives occurring in the system (unless we are to go in circles). Hence, solving for the first variable to be eliminated can be done in 2^n steps. Now the system has the correct form for the application of the Ackermann-rule. Application of this rule can increase the size of the system, for there may be multiple negative occurrences of p being substituted for. This substitution can result in no more than a squaring of the size of the system. The next variable is solved for in the system so obtained, and eliminated, leading to yet another possible increase in the size of the system, and so on. It follows that, given the right ordering of variables, each system can transformed into a pure one in 2^{n2^n} steps. There are at most $n!$ possible orderings of the variables, that is less than $2^{n \log n}$, and hence the total execution time of the algorithm bounded above by $2^n + 2^{n^2.2^n. \log n}$.

It follows that the time it takes SQEMA to reduce an input formula (or report failure) is bound above by a double exponential function of the size of the original input formula.

2.4 Correctness

In the previous two sections we respectively introduced the basic SQEMA-algorithm and gave some examples of how it can be used to compute (local) first-order frame correspondents for $\mathcal{L}$-formulae. We have, however, not yet shown that the algorithm is sound, viz. that the results produced by it are indeed (locally) equivalent on frames to the input formulae. This is what we do in this section. To that aim, we make the following definitions. Note that throughout this section the term *model* will be used to refer to models suitable for the interpretation of $\mathcal{L}_r^n$, i.e. models (W, R, V) where the valuation V also interprets nominals.

Definition 2.4.1 Let $\mathcal{M} = (\mathfrak{F}, V)$ and $\mathcal{M}' = (\mathfrak{F}, V')$ be two models over the same Kripke frame, and let PROP and NOM be sets of propositional variables and nominals, respectively. We say that $\mathcal{M}$ and $\mathcal{M}'$ are *(PROP, NOM)-related* if

1. $V'(p) = V(p)$ or $V'(p) = W - V(p)$ for all $p \in \mathsf{PROP}$, and
2. $V'(\mathbf{j}) = V(\mathbf{j})$ for all $\mathbf{j} \in \mathsf{NOM}$.

The next definition is intended to capture the type of equivalence which is preserved by the SQEMA-transformation rules.

Definition 2.4.2 Formulae $\varphi, \psi \in \mathcal{L}_r^n$ are *transformation equivalent* if, for every model $\mathcal{M} = (\mathfrak{F}, V)$ such that $\mathcal{M} \Vdash \varphi$ there exists a $(\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi), \mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi))$-related model $\mathcal{M} = (\mathfrak{F}, V')$ such that $\mathcal{M}' \Vdash \psi$, and vice versa. We will write $\varphi \equiv_{\text{trans}} \psi$ to indicate that φ and ψ are transformation equivalent.

Remark 2.4.3 Note that transformation equivalence is not a proper equivalence relation, since, in general, it need not be transitive. For example, $p \vee \Box\bot \equiv_{\text{trans}} q \vee \Diamond\Diamond\top$ and $q \vee \Diamond\Diamond\top \equiv_{\text{trans}} p \vee \Diamond\Diamond\top$, but $p \vee \Box\bot \not\equiv_{\text{trans}} p \vee \Diamond\Diamond\top$. However, we have the following version of transitivity: if $\varphi_1 \equiv_{\text{trans}} \varphi_2$, $\varphi_2 \equiv_{\text{trans}} \varphi_3$, and $\mathsf{AT}(\varphi_1) \cap \mathsf{AT}(\varphi_3) \subseteq \mathsf{AT}(\varphi_2)$, then $\varphi_1 \equiv_{\text{trans}} \varphi_3$.

The following proposition lists two useful properties of transformation equivalence:

Proposition 2.4.4 *Let $\varphi, \psi \in \mathcal{L}_r^n$. Then*

1. *$\varphi \equiv_{\text{trans}} \psi$ whenever $\varphi \equiv_{\text{sem}} \psi$ or $\varphi \equiv_{\text{mod}} \psi$;*

Moreover, for any frame $\mathfrak{F} = (W, R)$, points $w_1, \ldots, w_n \in W$ and nominals $\mathbf{i}_1, \ldots, \mathbf{i}_n \in$ $\mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi)$,

2. *if $\varphi \equiv_{\text{trans}} \psi$, then φ is $[\mathbf{i}_1 := w_1, \ldots, \mathbf{i}_n := w_n]$-satisfiable on $\mathfrak{F}$ if and only if ψ is $[\mathbf{i}_1 := w_1, \ldots, \mathbf{i}_n := w_n]$-satisfiable on $\mathfrak{F}$.*

The relationship of transformation equivalence with equivalence over general frames, specifically descriptive fames, will be considered in more detail in the next section.

Proposition 2.4.5 *If Sys' is a SQEMA-system obtained from a system Sys by application of SQEMA-transformation rules, then $\mathsf{Form}(\mathsf{Sys})$ and $\mathsf{Form}(\mathsf{Sys}')$ are transformation equivalent.*

Proof. Firstly, it is easy to see that each transformation rule preserves transformation equivalence. The case for the Ackermann-rule is justified by lemma 2.1.4. Secondly, the sequence of systems obtained satisfies the requirements for the limited version of transitivity (remark 2.4.3), since no eliminated variable ever reappears, nor does any nominal which is eliminated (e.g. by auxiliary rules), since the $\Diamond$-rule requires *new* nominals. QED

Theorem 2.4.6 (Correctness of SQEMA) *If SQEMA succeeds on an input formula $\varphi \in \mathcal{L}$, then the first-order formula returned is a local frame correspondent of φ.*

Proof. Suppose that SQEMA succeeds on $\varphi \in \mathcal{L}$. For simplicity, and without loss of generality, assume that the execution does not branch because of disjunctions. We may make this assumption since the conjunction of local first-order correspondents of modal formulae is a local first-order correspondent for the conjunction of those formulae.

Let $\mathfrak{F} = (W, R)$ be a Kripke frame and $w \in W$. Let $\mathsf{Sys}_0, \ldots, \mathsf{Sys}_r$ be the sequence of systems of equations produced by SQEMA when executed on φ. We define the *second-order translation of a system Sys_j*, $\mathrm{TR}(\mathsf{Sys}_j)$, to be the second-order formula $\exists\overline{P}\exists\overline{y}\forall x_0 \mathrm{ST}(\mathsf{Form}(\mathsf{Sys}_j), x_0)$, where $\overline{P}$ is the tuple of all predicate variables and $\overline{y}$ the tuple of all variables corresponding to nominals *other than* $\mathbf{i}$, occurring in $\mathrm{ST}(\mathsf{Form}(\mathsf{Sys}_j), x_0)$. Note that $y_{\mathbf{i}}$, corresponding to $\mathbf{i}$, is the only free variable in $\mathrm{TR}(\mathsf{Sys}_j)$, and that $\mathrm{TR}(\mathsf{Sys}_r)$ is $\exists\overline{y}\forall x_0 \mathrm{ST}(\mathsf{pure}(\varphi), x_0)$. Then
$\mathfrak{F}, w \Vdash \varphi$ iff
$\mathfrak{F} \models \forall\overline{P} ST(\varphi, x_0)[x_0 := w]$ iff
$\mathfrak{F} \models \forall\overline{P}\exists x_0 ST(\mathbf{i} \wedge \varphi, x_0)[y_{\mathbf{i}} := w]$ iff
$\mathfrak{F} \not\models \exists\overline{P}\forall x_0 ST(\neg\mathbf{i} \vee \neg\varphi, x_0)[y_{\mathbf{i}} := w]$, i.e. iff
$\mathfrak{F} \not\models \mathrm{TR}(\mathsf{Sys}_0)[y_{\mathbf{i}} := w]$.
Now, by propositions 2.4.4(2) and 2.4.5, we have that $\mathfrak{F} \not\models \mathrm{TR}(\mathsf{Sys}_0)[y_{\mathbf{i}} := w]$ if and only if $\mathfrak{F} \not\models \mathrm{TR}(Sys_r)[y_{\mathbf{i}} := w]$.

Hence we get that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathfrak{F} \not\models \exists\overline{y}\forall x_0 \mathrm{ST}(\mathsf{pure}(\varphi), x_0)[y_{\mathbf{i}} := w]$, i.e that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathfrak{F} \models \forall\overline{y}\exists x_0 \neg\mathrm{ST}(\mathsf{pure}(\varphi), x_0)[y_{\mathbf{i}} := w]$.

Hence $\forall\overline{y}\exists x_0 \neg\mathrm{ST}(\mathsf{pure}(\varphi), x_0)$ is a local first-order correspondent for φ, and exactly what SQEMA returns. Accordingly, $\forall y_{\mathbf{i}}\forall\overline{y}\exists x_0 \neg\mathrm{ST}(\mathsf{pure}(\varphi), x_0)$ is a global first-order correspondent of φ. QED

Corollary 2.4.7 *If* SQEMA *succeeds on an input formula* $\varphi \in \mathcal{L}$*, then, for any pointed frame* $(\mathfrak{F}, w)$*, it is the case that* $(\mathfrak{F}, w) \Vdash \varphi$ *if and only if* $\mathfrak{F} \Vdash \neg \mathsf{pure}(\varphi)[\mathbf{i} := w]$.

Remark 2.4.8 The proof of theorem 2.4.6 is a good illustration of the strategy employed by SQEMA, of simulating local conditions with global ones, via the use of nominals. Specifically, it exploits the simple observation that a formula φ is satisfiable at a point w in a frame $\mathfrak{F}$, if and only if $\mathbf{i} \to \varphi$ is *globally* satisfiable on $\mathfrak{F}$ with $\mathbf{i}$ denoting w. Working with global rather than local satisfiability is essential — not only is it needed for the correct application of (the modal version of) Ackermann's lemma, but it also allows the 'change of perspective' formalized in the $\Diamond$ and $\Box$-rules. For example, the $\Box$-rule allows us to take the claim $\mathcal{M} \Vdash \mathbf{j} \to \Box\psi$ (which can be seen as a claim about the state of affairs relative to the point named by $\mathbf{j}$), and replace it by the statement $\mathcal{M} \Vdash \Diamond^{-1}\mathbf{j} \to \psi$, which changes the perspective to the successors of the point named by $\mathbf{j}$.

2.5 Canonicity

As already mentioned in chapter 1, first-order definability and canonicity are properties that often go hand in hand for modal formulae. In this section we show that the formulae for which SQEMA computes first-order equivalents are no exception to this rule, viz. that they are all canonical.

This will be done by showing that the $\mathcal{L}$-input formulae on which SQEMA succeeds are d-persistent (recall definitions 0.1.1 and 1.2.8). We will therefore have to work with arbitrary (i.e. not necessarily reversive (subsection 0.1.2)) descriptive general frames, but since SQEMA introduces inverse modalities and nominals, it will be necessary to interpret $\mathcal{L}_r^n$-formulae over these frames. The details of this way of interpreting $\mathcal{L}_r^n$-formulae are spelled out in subsection 2.5.2. It is therefore important to bear in mind that, throughout this section, whenever the term 'descriptive general frame' is used, the general frame referred to need *not* be reversive.

2.5.1 Descriptive frames — a topological view

We will work in the basic modal language $\mathcal{L}$, but all definitions generalize to arbitrary similarity types. Our approach will be topological and similar to that of [SV89]. The topological notions used will be quite basic, and can be found in any standard topology text, e.g. [Wil04].

With every general frame $\mathfrak{F} = (W, R, \mathbb{W})$, we associate a topological space $(W, T(\mathfrak{F}))$, where $\mathbb{W}$ is taken as a base of clopen sets for the topology $T(\mathfrak{F})$. Let $\mathbf{C}(\mathbb{W})$ denote the set of sets closed with respect to $T(\mathfrak{F})$, i.e. $\mathbf{C}(\mathbb{W})$ is the set all intersections of members of $\mathbb{W}$. Once again we use the notations $\Diamond$, m_R and $\langle R \rangle$ (respectively $\Box$, l_R and $[R]$) interchangeably.

Let us reiterate the definition of a descriptive frame, now adding topological equivalents for the various clauses.

Definition 2.5.1 A general frame $\mathfrak{F} = (W, R, \mathbb{W})$ is said to be:

differentiated if for every $x, y \in W$, $x \neq y$, there exists $X \in \mathbb{W}$ such that $x \in X$ and $y \notin X$ (equivalently, if $T(\mathfrak{F})$ is Hausdorff);

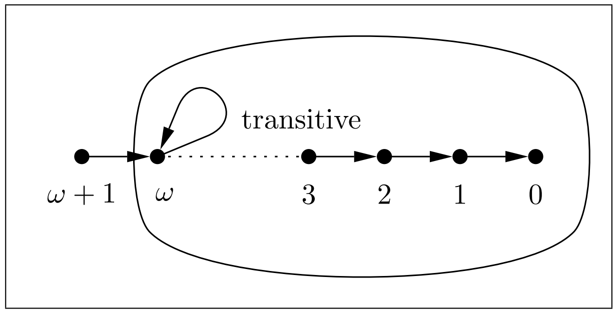

Figure 2.1: Example 2.5.2

tight if for all $x, y \in W$ it is the case that Rxy iff $x \in \bigcap\{\langle R\rangle(Y) \mid Y \in \mathbb{W}$ and $y \in Y\}$ (equivalently, if R is point-closed, i.e. R({x}) is closed for every $x \in W$);

compact if every family of admissible sets from $\mathbb{W}$ with the finite intersection property (FIP) has non empty intersection (equivalently, if $T(\mathfrak{F})$ is compact);

refined if it is differentiated and tight;

descriptive it is refined and compact.

Note that for such frames the algebra of admissible sets $\mathbb{W}$ need not be closed under the $m_{R^{-1}}$-operator (usually written simply as $\Diamond^{-1}$), as illustrated in example 2.5.2, below. However, the algebras of descriptive frames for the reversive langauge $\mathcal{L}_r$ *will* be closed under this operator. For the remainder of this section, whenever talking about a descriptive frame we will always mean a descriptive frame for the basic modal language.

Example 2.5.2 Let $\mathfrak{F} = (W, R, \mathbb{W})$ be the general frame with underlying Kripke frame pictured in figure 2.5.1. Note that ω is reflexive while all other points are irreflexive. Further, the only successor of $\omega+1$ is ω, while the relation in the submodel generated by ω is transitive. Let $\mathbb{W} = \{X_1 \cup X_2 \cup X_3 \mid X_i \in \mathbb{X}_i, i = 1, 2, 3\}$, where $\mathbb{X}_1$ contains all finite (possibly empty) sets of natural numbers, $\mathbb{X}_2$ contains $\emptyset$ and all sets of the form $\{x \in W \mid n \leq x \leq \omega\}$ for all $n \in \omega$, and $\mathbb{X}_3 = \{\emptyset, \{\omega+1\}\}$. It is not difficult to check that $\mathfrak{F}$ is descriptive. (This general frame is given in example 8.52 in [CZ97].)

Now, $\{\omega+1\}$ is an admissible set, but $\Diamond^{-1}(\{\omega+1\}) = \{\omega\}$, which is not admissible. Hence the algebra of admissible sets is not closed under the $\Diamond^{-1}$ operator. $\triangleleft$

2.5.2 Augmented models

When wishing to interpret $\mathcal{L}_r^n$ formulae on descriptive frames[2], we have to specify how nominals are to be interpreted: will their valuations range over all singletons, only admissible singletons, or some other subset of the domain of $\mathfrak{F}$? In this section we opt to let them range over all singletons, and to formalize this we make the following definition.

[2]Recall that these frames need not be reversive.

Definition 2.5.3 Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a descriptive frame. An *augmented valuation on* $\mathfrak{F}$ is a function $V : \mathsf{AT} \to 2^W$ such that $V(p) \in \mathbb{W}$ for every propositional variable p and, for every nominal $\mathbf{j}$, $V(\mathbf{j}) = \{w\}$ for some $w \in W$. An *augmented model* based on $\mathfrak{F}$ is a model $(\mathfrak{F}, V)$ with V an augmented valuation on $\mathfrak{F}$. A $\mathcal{L}_r^n$-formula φ is *augmentedly satisfiable* at a point w in $\mathfrak{F}$ if there exists an augmented model $(\mathfrak{F}, V)$ based on $\mathfrak{F}$ such that $((\mathfrak{F}, V), w) \Vdash \varphi$. Local and global augmented satisfiability and validity are defined analogously. We will write $(\mathfrak{F}, w) \Vdash^{aug} \varphi$ if φ is augmentedly valid at w in $\mathfrak{F}$.

Definition 2.5.4 An $\mathcal{L}_r^n$-formula is locally *locally ad-persistent* if, for all descriptive frames $\mathfrak{F} = (W, R, \mathbb{W})$ and points $w \in W$, it is the case that $(\mathfrak{F}, w) \Vdash^{aug} \varphi$ only if $(\mathfrak{F}_\sharp, w) \Vdash \varphi$. Similarly, a $\mathcal{L}_r^n$-formula is *ad-persistent* if, for all descriptive frames $\mathfrak{F} = (W, R, \mathbb{W})$, it is the case that $\mathfrak{F} \Vdash^{aug} \varphi$ only if $\mathfrak{F}_\sharp \Vdash \varphi$.

The next two propositions are direct consequences of the above definitions.

Proposition 2.5.5 *Every $\mathcal{L}$-formula is (locally) ad-persistent if and only if it is (locally) d-persistent.*

Proposition 2.5.6 *Every pure $\mathcal{L}_r^n$-formula is (locally) ad-persistent.*

Adapting definition 2.4.2 to the case of $\mathcal{L}_r^n$ -formulae interpreted in augmented models, we obtain:

Definition 2.5.7 Formulae $\varphi, \psi \in \mathcal{L}_r^n$ are *ad-transformation equivalent* if, for every augmented model $\mathcal{M} = (\mathfrak{F}, V)$ based on a descriptive frame $\mathfrak{F}$, such that $\mathcal{M} \Vdash \varphi$ there exists a $(\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi), \mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi))$-related augmented model $\mathcal{M} = (\mathfrak{F}, V')$ based on $\mathfrak{F}$ such that $\mathcal{M}' \Vdash \psi$, and vice versa. We will write $\varphi \equiv_{\text{trans}}^{ad} \varphi$ if φ and ψ are ad-transformation equivalent.

Analogously to clause 2 of proposition 2.4.4 we have:

Proposition 2.5.8 *Let $\varphi, \psi \in \mathcal{L}_r^n$. Then, for any descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$, points $w_1, \ldots, w_n \in W$ and nominals $\mathbf{i}_1, \ldots, \mathbf{i}_n \in \mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi)$, if $\varphi \equiv_{\text{trans}}^{ad} \psi$, then φ is globally $[\mathbf{i}_1 := w_1, \ldots, \mathbf{i}_n := w_n]$-satisfiable in some augmented model based on $\mathfrak{F}$ if and only if ψ is so satisfiable.*

The desired canonicity result will follow easily, once we have established an analogue of proposition 2.4.5 for ad-transformation equivalence. Most of the work in the rest of this section is done for the sake of establishing such a result.

2.5.3 $\mathcal{L}_r^n$-formulae as operators on descriptive frames

In this section we prove some topological properties of $\mathcal{L}_r^n$-formulae, regarded as operators on descriptive frames (as explained in subsection 0.1.2). These properties will enables us to prove lemma 2.5.20, a version of Ackermann's lemma for descriptive frames.

Definition 2.5.9 Let $\gamma = \gamma(p_1, \ldots, p_n, \mathbf{i}_1, \ldots, \mathbf{i}_m)$ be a $\mathcal{L}_r^n$-formula, with $\mathsf{PROP}(\gamma) = \{p_1, \ldots, p_n\}$ and $\mathsf{NOM}(\gamma) = \{\mathbf{i}_1, \ldots, \mathbf{i}_m\}$. We say γ is a *closed operator on descriptive frames* if, for every descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$, any $P_1, \ldots, P_n \in \mathbf{C}(\mathbb{W})$ and any $x_1, \ldots, x_m \in W$, it is the case that

$$\gamma(P_1, \ldots, P_n, \{x_1\}, \ldots, \{x_m\}) \in \mathbf{C}(\mathbb{W}),$$

i.e. when applied to closed sets in a descriptive frame it produces a closed set.

We say γ is a *closed formula on descriptive frames* if, for every descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$, any $P_1, \ldots, P_n \in \mathbb{W}$ and any $x_1, \ldots, x_m \in W$, it is the case that

$$\gamma(P_1, \ldots, P_n, \{x_1\}, \ldots, \{x_m\}) \in \mathbf{C}(\mathbb{W}),$$

i.e. whenever applied to *admissible* sets in any descriptive frame, and with nominals allowed to range over arbitrary singletons, it produces a closed set.

Thus, if a formula is a closed operator, then it is a closed formula, but not necessarily vice versa.

Similarly, a $\mathcal{L}_r^n$-formula is an *open operator on descriptive frames* if whenever applied to open sets in a descriptive frame (with nominals ranging over singletons) it produces an open set. It is an *open formula on descriptive frames* if, whenever applied to admissible sets (and with nominals ranging over singletons), it produces an open set.

Note that the operators $\Diamond$ and $\Diamond^{-1}$ distribute over arbitrary unions, while $\Box$ and $\Box^{-1}$ distribute over arbitrary intersections. Since every closed set can be obtained as the intersection of admissible sets and each open set as the union of admissible sets, and $\Diamond$ and $\Box$ applied to admissible sets yield admissible sets, it follows that $\Diamond p$ is an open operator and $\Box p$ is a closed operator, and this holds even for arbitrary general frames. The proof of the next lemma, originally from [Esa74], can also be found in [SV89] or [BdRV01].

Lemma 2.5.10 (Esakia's lemma for $\Diamond$) *Let $\mathfrak{F}$ be a descriptive frame. Then for any downward directed family of nonempty closed sets $\{C_i \mid i \in I\}$ from $\mathbf{C}(\mathfrak{F})$, it is the case that $\Diamond \bigcap_{i \in I} C_i = \bigcap_{i \in I} \Diamond C_i$.*

Once again using the fact that closed set can be obtained as the intersection of admissible sets, we obtain the following corollary.

Corollary 2.5.11 *$\Diamond p$ is both a closed and an open operator on descriptive frames.*

By the duality of $\Box$ and $\Diamond$ we also immediately have:

Corollary 2.5.12 *$\Box p$ is both a closed and an open operator on descriptive frames.*

Lemma 2.5.13 (Esakia's lemma for $\Diamond^{-1}$ on descriptive frames) *Let $\mathfrak{F}$ be a descriptive frame. Then $\Diamond^{-1} \bigcap_{i \in I} C_i = \bigcap_{i \in I} \Diamond^{-1} C_i$ for any downwards directed family of nonempty closed sets $\{C_i \mid i \in I\}$ from $\mathbf{C}(\mathfrak{F})$.*

Proof. The inclusion $\Diamond^{-1} \bigcap_{i \in I} C_i \subseteq \bigcap_{i \in I} \Diamond^{-1} C_i$ is trivial, so suppose that $x_0 \notin \Diamond^{-1} \bigcap_{i \in I} C_i$, i.e. $\langle R \rangle(x_0) \cap \bigcap_{i \in I} C_i = \emptyset$. Now $\langle R \rangle(x_0)$ is closed by corollary 2.5.11 and the fact that singletons are closed in descriptive frames. Hence $\{\langle R \rangle(x_0)\} \cup \{C_i \mid i \in I\}$ is a family of closed subsets with empty intersection which, by compactness, cannot have the FIP. Thus there is a finite subfamily $\{C_1, \ldots, C_n\} \subseteq \{C_i \mid i \in I\}$ such that $\langle R \rangle(x_0) \cap C_1 \cap \cdots \cap C_n = \emptyset$. Since $\{C_i \mid i \in I\}$ is downwards directed, it follows that there that there exists a $C \in \{C_i \mid i \in I\}$ such that $C \subseteq \bigcap\{C_1, \ldots, C_n\}$ and $\langle R \rangle(x_0) \cap C = \emptyset$. But then $x_0 \notin \Diamond^{-1} C$, and hence $x_0 \notin \bigcap_{i \in I} \Diamond^{-1} C_i$. QED

Note that, because $\Diamond^{-1}$ need nor produce an admissible sets when applied to one, we cannot obtain the closedness of $\Diamond^{-1} p$ as an operator on descriptive frames from lemma 2.5.13 in the same way that we obtained corollary 2.5.11 from lemma 2.5.10. The proof of the following lemma is adapted from [GV06].

Lemma 2.5.14 *$\Diamond^{-1} p$ is a closed operator on descriptive frames.*

Proof. Let $\mathfrak{F} = \langle W, R, \mathbb{W} \rangle$ be a descriptive frame. We will show that for any closed subset $A \subseteq W$ it is the case that $\Diamond^{-1}(A) = \bigcap\{B \in \mathbb{W} \mid \Diamond^{-1} A \subseteq B\}$. Note that $\Diamond^{-1}(A) = R(A)$, where $R(A)$ denotes the set of all R-successors of elements of A. The inclusion from left to right is trivial. In order to prove the right-to-left inclusion, suppose that $x_0 \notin \Diamond^{-1}(A)$, i.e. for all $y \in A$ it is not the case that Ryx_0. By the point-closedness of R we have that $R(y) = \bigcap\{B \in \mathbb{W} \mid y \in \Box B\}$. Then for each $y \in A$ there must exist a $B^y \in \mathbb{W}$ such that $y \in \Box B^y$ and $x_0 \notin B^y$, and hence $A \subseteq \bigcup\{\Box B^y \mid y \in A\}$. Therefore $\{\Box B^y \mid y \in A\}$ is an open cover of the closed set A, so by compactness there exists a finite subcover $\Box B_1, \ldots, \Box B_n$. Then $A \subseteq \Box B_1 \cup \cdots \cup \Box B_n$ and $x_0 \notin B_i, 1 \leq i \leq n$. Since $\Diamond^{-1}$ distributes over arbitrary unions, we then have $\Diamond^{-1} A \subseteq \Diamond^{-1} \Box B_1 \cup \cdots \cup \Diamond^{-1} \Box B_n$. And, since for any $X \subseteq W$, $\Diamond^{-1} \Box X \subseteq X$, we have $\Diamond^{-1} A \subseteq B_1 \cup \cdots \cup B_n$. So we have found an admissible set containing $\Diamond^{-1} A$, not containing x_0, and hence $x_0 \notin \bigcap\{B \in \mathbb{W} \mid \Diamond^{-1} A \subseteq B\}$, proving the inclusion and the lemma. QED

Corollary 2.5.15 *$\Box^{-1} p$ is an open operator on descriptive frames.*

Proof. By the duality of $\Diamond^{-1}$ and $\Box^{-1}$. QED

Definition 2.5.16 A formula $\varphi \in \mathcal{L}_r^n$ is *syntactically closed* if all occurrences of nominals and $\Diamond^{-1}$ in φ are positive, and all occurrences of $\Box^{-1}$ in φ are negative or, equivalently, when written in negation normal form, φ is positive in all nominals and contains no occurrences of $\Box^{-1}$.

A formula $\varphi \in \mathcal{L}_r^n$ is *syntactically open* if all occurrences of $\Diamond^{-1}$ and nominals in φ are negative, and all occurrences of $\Box^{-1}$ in φ are positive or, equivalently, when written in negation normal form, φ is negative in all nominals and contains no occurrences of $\Diamond^{-1}$.

Clearly $\neg$ maps syntactically open formulae to syntactically closed formulae, and vice versa.

Lemma 2.5.17 *Every syntactically closed $\mathcal{L}_r^n$-formula is a closed formula on descriptive frames, and every syntactically open $\mathcal{L}_r^n$-formula is an open formula on descriptive frames.*

Proof. By structural induction on syntactically open and closed formulae, written in negation normal form, using corollaries 2.5.11, 2.5.12 and 2.5.15 and lemma 2.5.14, as well as the fact that all singletons are closed in descriptive frames. QED

The next lemma will be used in the proof of Esakia's lemma for syntactically closed formulae:

Lemma 2.5.18 *Let $\varphi(q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots \mathbf{i}_m) \in \mathcal{L}_r^n$ be a syntactically closed formula which is positive in p and with $\mathsf{PROP}(\varphi) = \{q_1, \ldots, q_n, p\}$ and $\mathsf{NOM}(\varphi) = \{\mathbf{i}_1, \ldots \mathbf{i}_m\}$. Then for all descriptive fames $\mathfrak{F} = (W, R, \mathbb{W})$, and all $Q_1, \ldots, Q_n \in \mathbb{W}$, $x_1, \ldots, x_m \in W$, and $C \in \mathbf{C}(\mathbb{W})$, it is the case that $\varphi(Q_1, \ldots, Q_n, C, \{x_1\}, \ldots, \{x_m\})$ is closed in $T(\mathfrak{F})$.*

Proof. We assume that φ is written in negation-normal form, and hence that $\Box^{-1}$ does not occur. We proceed by induction on φ. If φ is $\top$, $\bot$ or one of $q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots \mathbf{i}_m$ it is clear that $\varphi(Q_1, \ldots, Q_n, \{x_1\}, \ldots, \{x_m\}, C)$ is a closed set. This is also the case if φ is the negation of a propositional variable from $q_1, \ldots, q_n$. The cases when φ is $\neg p$ or $\neg \mathbf{i}_j$ do not occur.

The inductive cases for $\wedge$ and $\vee$ follow since the finite unions and intersections of closed sets are closed. The cases for $\Diamond$ and $\Diamond^{-1}$ follow from corollary 2.5.11 and 2.5.14, respectively. Lastly, the case for $\Box$ follows from corollary 2.5.12. QED

Lemma 2.5.19 (Esakia's lemma for Syntactically Closed Formulae) *Let $\varphi(q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots \mathbf{i}_m) \in \mathcal{L}_r^n$ be a syntactically closed formula with $\mathsf{PROP}(\varphi) = \{q_1, \ldots, q_n, p\}$ and $\mathsf{NOM}(\varphi) = \{\mathbf{i}_1, \ldots \mathbf{i}_m\}$ which is positive in p. Then for every descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$, all $Q_1, \ldots, Q_n \in \mathbb{W}$, $x_1, \ldots, x_m \in W$ and any downwards directed family of closed sets $\{C_i \mid i \in I\}$ from $\mathbf{C}(\mathfrak{F})$, it is the case that*

$$\varphi(Q_1, \ldots, Q_n, \bigcap_{i \in I} C_i, \{x_1\}, \ldots, \{x_m\}) = \bigcap_{i \in I} \varphi(Q_1, \ldots, Q_n, C_i, \{x_1\}, \ldots, \{x_m\}).$$

Proof. The proof is by induction on φ. For brevity we will omit the parameters $Q_1, \ldots, Q_n$, $\{x_1\}, \ldots, \{x_m\}$ when writing (sub)formulae. We assume that formulae are written in negation-normal form, and hence that $\Box^{-1}$ does not occur. The cases when φ is $\bot$, $\top$ or among $q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots \mathbf{i}_m$ are trivial, as are the cases when φ is the negation of a propositional variable among $q_1, \ldots, q_n$. The cases when φ is $\neg p$ or $\neg \mathbf{i}_j$ do not occur. The inductive step in the case when φ is of the form $\gamma_1 \wedge \gamma_2$ is also trivial.

Suppose φ is of the form $\gamma_1 \vee \gamma_2$. We have to show that $\gamma_1(\bigcap_{i \in I} C_i) \cup \gamma_2(\bigcap_{i \in I} C_i) = \bigcap_{i \in I}(\gamma_1(C_i) \cup \gamma_2(C_i))$. The interesting inclusion is from right to left, so assume that $x_0 \notin \gamma_1(\bigcap_{i \in I} C_i) \cup \gamma_2(\bigcap_{i \in I} C_i)$, i.e $x_0 \notin \bigcap_{i \in I} \gamma_1(C_i) \cup \bigcap_{i \in I} \gamma_2(C_i)$, by the induction hypothesis. Thus there exists $C_1, C_2 \in \{C_i \mid i \in I\}$ such that $x_0 \notin \gamma_1(C_1)$ and $x_0 \notin \gamma_2(C_2)$. By the downward directedness of $\{C_i \mid i \in I\}$ there is a $C \in \{C_i \mid i \in I\}$ such that $C \subseteq C_1 \cap C_2$. Thus, since γ_1 and γ_2 are positive and hence upwards monotone in p, it follows that $x_0 \notin \gamma_1(C)$ and $x_0 \notin \gamma_2(C)$, and hence that $x_0 \notin \bigcap_{i \in I}(\gamma_1(C_i) \cup \gamma_2(C_i))$.

Suppose φ is of the form $\Diamond\gamma$. We have to show that $\Diamond\gamma(\bigcap_{i \in I} C_i) = \bigcap_{i \in I} \Diamond\gamma(C_i)$. By the inductive hypothesis we have $\Diamond\gamma(\bigcap_{i \in I} C_i) = \Diamond \bigcap_{i \in I} \gamma(C_i)$. If $\gamma(C_i) = \emptyset$ for some C_i, then $\Diamond \bigcap_{i \in I} \gamma(C_i) = \emptyset = \bigcap_{i \in I} \Diamond\gamma(C_i)$, so we may assume that $\gamma(C_i) \neq \emptyset$ for all $i \in I$. Then, by lemma 2.5.18, $\{\gamma(C_i) \mid i \in I\}$ is a family of non-empty closed sets. Moreover, $\{\gamma(C_i) \mid i \in I\}$ is downwards directed. For, consider any finite number of members of $\{\gamma(C_i) \mid i \in I\}$, $\gamma(C_1), \ldots, \gamma(C_n)$, say. Then there is a $C \in \{C_i \mid i \in I\}$ such that $C \subseteq \bigcap_{i=1}^n C_i$. But then $\gamma(C) \in \{\gamma(C_i) \mid i \in I\}$ and $\gamma(C) \subseteq \bigcap_{i=1}^n \gamma(C_i)$ by the upwards monotonicity of γ in p. Now we may apply lemma 2.5.10 and conclude that $\Diamond \bigcap_{i\in I} \gamma(C_i) = \bigcap_{i\in I} \Diamond\gamma C_i)$.

The case when φ is of the form $\Diamond^{-1}\gamma$ is verbatim the same the previous case, except that we appeal to lemma 2.5.13 rather than lemma 2.5.10 in the last step.

Lastly consider the case when φ is of the form $\Box\gamma$. This follows by the inductive hypothesis and the fact that $\Box$ distributes over arbitrary intersections of subsets of W. QED

The next lemma is needed for the following reason: in order to show that SQEMA preserves ad-transformation equivalence, we will need a version of Ackermann's lemma that is true of $\mathcal{L}_r^n$-formulae when interpreted over descriptive frames (for the basic modal language $\mathcal{L}$). As already noted, the extension of an $\mathcal{L}_r^n$-formula in an augmented model based on such a general frame need not be an admissible set in that general frame. This creates and obvious impediment for the proof of one direction of the equivalence in the modal Ackermann's lemma, as the interpretations of propositional variables *must* be admissible sets. We can push it through, however, at the price of additional restrictions on formulae in terms of syntactic openness and closedness.

Lemma 2.5.20 (Restricted Version of Ackermann's lemma for Descriptive Frames) *Suppose $A \in \mathcal{L}_r^n$ is a syntactically closed formula and $B(p) \in \mathcal{L}_r^n$ is a syntactically open formula which is negative in p. Then*

$$((A \to p) \wedge B(p)) \equiv_{\text{trans}}^{ad} B(A/p).$$

Proof. Let $A(q_1, \ldots, q_n, \mathbf{i}_1, \ldots, \mathbf{i}_m)$ and $B(q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots, \mathbf{i}_m)$ be as in the formulation of the lemma, with $\mathsf{PROP}(A) \subseteq \{q_1, \ldots, q_n\}$, $\mathsf{PROP}(B) \subseteq \{q_1, \ldots, q_n, p\}$ and $\mathsf{NOM}(A)$, $\mathsf{NOM}(B) \subseteq \{\mathbf{i}_1, \ldots, \mathbf{i}_m\}$. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a descriptive frame, and $Q_1, \ldots, Q_n \in \mathbb{W}$ and $x_1, \ldots, x_m \in W$. We will that

$$B(Q_1, \ldots, Q_n, A(Q_1, \ldots, Q_n, \{x_1\}, \ldots, \{x_m\}), \{x_1\}, \ldots, \{x_m\}) = W$$

if and only if there is a $P \in \mathbb{W}$ such that

$$A(Q_1, \ldots, Q_n, \{x_1\}, \ldots, \{x_m\}) \subseteq P \text{ and } B(Q_1, \ldots, Q_n, P, \{x_1\}, \ldots, \{x_m\}) = W,$$

which is sufficient for the ad-transformation equivalence of (A $\to B(p)$) and $B(A/p)$.

For the sake of brevity we will omit the parameters $Q_1, \ldots, Q_n, \{x_1\}, \ldots, \{x_m\}$ in the rest of the proof, and simply write A, $B(P)$, etc. The implication from bottom to top follows by the downwards monotonicity of B in p.

For the converse, assume that $B(A) = W$. Let $B'(p)$ be the negation of $B(p)$ written in negation normal form. Then $B'(p)$ is a syntactically closed formula, positive in p, and $B'(A) = \emptyset$. We need to find an admissible set $P \in \mathbb{W}$ such that $A \subseteq P$ and $B'(P) = \emptyset$. Since A is a syntactically closed formula, it follows by lemma 2.5.17 that A is a closed subset of W and hence that $A = \bigcap\{C \in \mathbb{W} \mid A \subseteq C\}$. Hence $\emptyset = B'(A) = B'(\bigcap\{C \in \mathbb{W} \mid$

$A \subseteq C\}) = \bigcap\{B'(C) \mid C \in \mathbb{W}$ and $A \subseteq C\}$, by lemma 2.5.19. Again by lemma 2.5.17, $\{B'(C) \mid C \in \mathbb{W}, A \subseteq C\}$ is a family of closed sets with empty intersection. Hence, by compactness, there must be a finite subfamily, $C_1, \ldots, C_n$ say, such that $\bigcap_{i=1}^{n} B'(C_i) = \emptyset$. But then $C = \bigcap_{i=1}^{n} C_i$ is an admissible set containing A, and $B'(C) = \emptyset$, i.e $B(C) = W$. Hence we can choose $P = C$. QED

2.5.4 Proving canonicity

Equipped with lemma 2.5.20, we are now almost ready to prove theorem 2.5.23, which asserts the canonicity of all $\mathcal{L}$-formulae on which SQEMA succeeds. First, however, we need two preliminary lemmas. Diamond-link sequents were defined on page 45.

Lemma 2.5.21 *During the entire (successful or unsuccessful) execution of SQEMA on any $\mathcal{L}$ input formula, all antecedents of all non-diamond-link sequents are syntactically closed formulae, while all consequents of all non-diamond-link sequents are syntactically open.*

Proof. We follow any one disjunctive branch of the execution, proceeding by induction on the application of transformation rules. The initial system is of the form $\|\mathbf{i} \Rightarrow \psi$, where $\psi \in \mathcal{L}$. For this system the conditions of the lemma hold, since $\mathbf{i}$ is syntactically closed and all $\mathcal{L}$-formulae are both syntactically closed and open. Now suppose that in the process of the execution we have reached a system satisfying the conditions of the lemma, viz. all antecedents of all non-diamond-link sequents in the system are syntactically closed, while all consequents of all non-diamond-link sequents in the system are syntactically open. It is straightforward to check that the application of any transformation rule to this system will preserve these conditions. In the particular case when the Ackermann-rule is applied, we note the following: (i) the diamond-link sequents in the system contain no propositional variables, and are hence essentially disregarded by in any application of the Ackermann-rule; (ii) by the inductive hypothesis the disjunction of antecedents which is substituted for the variable being eliminated is syntactically closed; (iii) substituting a syntactically closed formula for negative occurrences of a variable in a syntactically open formula yields a syntactically open formula. QED

We are now able to prove the next, crucial proposition. It is an analogue of proposition 2.4.5 for ad-transformation equivalence.

Proposition 2.5.22 *Let Sys be a system obtained during the execution of SQEMA on an $\mathcal{L}$-formula, and let Sys$'$ be obtained from Sys by the application of SQEMA-transformation rules. Then Form(Sys) $\equiv^{ad}_{\text{trans}}$ Form(Sys$'$).*

Proof. If we can verify that each transformation rule preserves ad-transformation equivalence, the result will follow from the limited version of transitivity satisfied by transformation equivalence (remark 2.4.3). We only verify that the Ackermann-rule — the cases for the other rules are trivial. To that end, suppose that Sys_2 is obtained from Sys_1 by the application of the Ackermann-rule. Then, by lemma 2.5.21, $\mathsf{Form}(\mathsf{Sys}_1)$ is of the form

$$\bigwedge_{j=1}^{l}(\neg\alpha_j \vee p) \wedge \bigwedge_{j=1}^{m}(\beta_j) \wedge \bigwedge_{j=1}^{n}\gamma_j$$

for some propositional variable p, syntactically closed formulae $\alpha_1, \ldots, \alpha_l$, syntactically open formulae $\beta_1, \ldots, \beta_m$ negative in p, and pure formulae $\gamma_1, \ldots, \gamma_n$ corresponding to diamond-link sequents. Hence $\mathsf{Form}(\mathsf{Sys}_2)$ is of the form

$$\bigwedge_{j=1}^{m}(\beta'_j) \wedge \bigwedge_{j=1}^{n}\gamma_j,$$

where each β'_j is obtained from β_j by substituting $\bigvee_{j=1}^{l}\alpha_j$ for all occurrences of p. The proof is complete once we appeal to lemma 2.5.20. QED

Theorem 2.5.23 *If* SQEMA *succeeds on a formula $\varphi \in \mathcal{L}$, then φ is locally d-persistent and hence canonical.*

Proof. Suppose that SQEMA succeeds on $\varphi \in \mathcal{L}$. Further, for simplicity and without loss of generality, assume that the execution does not branch because of disjunctions. We may make this assumption since a conjunction of d-persistent formulae is d-persistent.

Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a descriptive frame and $w \in W$. Then $(\mathfrak{F}, w) \Vdash \varphi$ iff $(\mathbf{i} \rightarrow \neg\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$ iff (by propositions 2.5.22 and 2.5.8) $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$.

By proposition 2.5.6, the latter is the case iff $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}_\sharp$, which in turn, by corollary 2.4.7 is the case if and only if $(\mathfrak{F}_\sharp, w) \Vdash \varphi$.

Hence we have established that $(\mathfrak{F}, w) \Vdash \varphi$ iff $(\mathfrak{F}_\sharp, w) \Vdash \varphi$, whence the local d-persistence of φ. QED

2.6 Some completeness results for syntactic classes

In this section we establish completeness results for SQEMA with respect to some syntactically specified classes of elementary and canonical formulae. By 'completeness' we mean that SQEMA succeeds in computing first-order equivalents for (and simultaneously proving the canonicity of) all members of these classes.

2.6.1 Sahlqvist and Sahlqvist–van Benthem formulae

We show that SQEMA succeeds in computing first-order equivalents for all Sahlqvist–van Benthem formulae, from which its completeness for the subclass of Sahlqvist formulae immediately follows. In the light of theorem 2.5.23, the canonicity of the members of theses two syntactic classes then follows as a corollary. First, a few preliminary notions.

Definition 2.6.1 Call a system Sys a *simple dual Sahlqvist–van Benthem system*, or an SDSBS for short, if each sequent in Sys has the from $\mathbf{j} \Rightarrow \psi$, where $\mathbf{j}$ is a nominal, and ψ is an $\mathcal{L}_r^n$-formula such that, in ψ:

(SDSBS1) no positive occurrence of a propositional variable is in a subformula of the form $\Diamond\gamma$, which is in the scope of a $\Box$,

(SDSBS2) no positive occurrence of a propositional variable is in the scope of a disjunction, and

(SDSBS3) no occurrence of a propositional variable is in the scope of an inverse box or an inverse diamond.

The sequents in an SDSBS will be refereed to as SDSBS-sequents.

Lemma 2.6.2 *Let p be a be a propositional variable occurring both positively and negatively in an SDSBS Sys. Then p may be eliminated from Sys by the application of the $\Diamond$-rule, $\Box$-rule, $\wedge$-rule, and Ackermann-rule, yielding an SDSBS Sys'.*

Proof. Let Sys be as in the formulation of the lemma. This means that the positive occurrences of p in the sequents of Sys are at most in the scope of $\Diamond$'s, $\Box$'s and $\wedge$'s with, moreover, no positive occurrence of p in the scope of a $\Diamond$ which is in the scope of a $\Box$. Note that application of the $\Diamond$ and $\wedge$-rules to an SDSBS again yields an SDSBS. Hence the $\Diamond$ and $\wedge$-rules can be applied to Sys until it has been transformed into an SDSBS Sys_1, in which every sequent in which p appears positively is has the form $\mathbf{j} \Rightarrow \gamma$, where $\mathbf{j}$ is a nominal and, in γ, all positive occurrences of p are at most in the scope of $\Box$'s and $\wedge$'s. Further application of $\Box$ and $\wedge$-rules yield a system Sys_2, where every sequent is either an SDBS-sequent or one of the form $\delta \Rightarrow p$ with δ a pure $\mathcal{L}_r^n$-formula. So Sys_2 need not in general be an SDSBS. Call the system obtained by applying to Sys_2 the Ackermann-rule with respect to p, Sys_3. Now Sys_3 is an SDSBS not containing any occurrence of the variable p. Indeed, when the Ackermann rule is applied to Sys_2, all non-SDSBS-sequents are eliminated, and a pure $\mathcal{L}_r^n$-formula is substituted for the (negative) occurrences of p in the SDSBS-sequents of Sys_2, yielding SDSBS-sequents. Sys_3 is then the desired system Sys'. QED

Theorem 2.6.3 *SQEMA succeeds on every Sahlqvist–van Benthem formula.*

Proof. Let φ be a Sahlqvist–van Benthem formula. Preprocessing negates φ and transforms it, by importing negation and bubbling up disjunctions, into a formula $\bigvee_{i=1}^{n} \varphi_i$, where in each φ_i each disjunction occurrence is in the scope of $\Box$. It is easy to see that, moreover, for each φ_i, the system $\|\mathbf{i} \Rightarrow \varphi_i$, if not already an SDSBS, may be transformed into an SDSBS by applying the polarity switching rule. The theorem now follows by induction on the number of occurring propositional variables, and appealing to lemma 2.6.2. QED

Corollary 2.6.4 *SQEMA succeeds on every Sahlqvist formula.*

Corollary 2.6.5 *Every Sahlqvist–van Benthem formula and every Sahlqvist formula is locally elementary, locally d-persistent, and hence canonical.*

2.6.2 Monadic inductive formulae

Now we extend the results of the previous section to monadic inductive formulae (recall definition 1.3.16). We now introduce a slight liberalization of this definition which allows for the restricted use of $\Box^{-1}$, $\Diamond^{-1}$ and nominals. Recall that a monadic box formula in $\mathcal{L}$ with head p is any formula of the form

$$A_0 \to \Box_1(A_1 \to \ldots \Box_n(A_n \to p)\ldots),$$

where $\Box_1, \ldots, \Box_n$ are finite, possibly empty, sequences of $\Box$'s, and $A_1, \ldots, A_n$ are positive formulae. For the purposes of this subsection we will mean by an $\mathcal{L}_r^n$*-liberalized box formula* a formula of the form

$$A_0 \vee \Box_1(A_1 \vee \ldots \Box_n(A_n \vee p)\ldots),$$

where $\Box_1, \ldots, \Box_n$ are as before, and $A_1, \ldots, A_n$ are any negative $\mathcal{L}_r^n$-formulae.

The $\mathcal{L}_r^n$*-liberalized monadic inductive formulae*, are then formulae built up from $\bot$, $\top$, negated $\mathcal{L}_r^n$-liberalized box formula and positive $\mathcal{L}_r^n$-formulae, using $\wedge$, $\vee$ and $\Diamond$, and which have acyclic dependency digraphs.

The abbreviation NegMIF will be used for the negation of an $\mathcal{L}_r^n$-liberalized monadic inductive formula in negation normal form. Note that the class of NegMIF's consists precisely of those formulae built from $\top$, $\bot$, negative $\mathcal{L}_r^n$-formulae and $\mathcal{L}_r^n$-liberalized box-formulae using $\wedge$, $\vee$, and $\Diamond$, which have acyclic dependency digraphs. The abbreviation NegMIF* will be used for NegMIF's built up without the use of disjunction, i.e. the class of all formulae of the language built from $\top$, $\bot$, negative $\mathcal{L}_r^n$-formulae and $\mathcal{L}_r^n$-liberalized box-formulae using $\wedge$ and $\Diamond$, which have acyclic dependency digraphs. Note that in NegMIF's and NegMIF*'s no positive occurrences of propositional variables are in the scope of $\Diamond^{-1}$ or $\Box^{-1}$.

Definition 2.6.6 Call a SQEMA-system a NegMIF-*system* (NegMIF*-*system*) if it has the form

$$\left\|\begin{array}{l} \mathbf{i}_1 \Rightarrow \alpha_1 \\ \vdots \\ \mathbf{i}_n \Rightarrow \alpha_n \end{array}\right. ,$$

where

(NMS1) each $\mathbf{i}_i$ is a nominal,

(NMS2) the formula $\alpha_1 \wedge \ldots \wedge \alpha_n$, obtained by taking the conjunction of all consequents of sequents in the system, is a NegMIF (NegMIF*).

Lemma 2.6.7 *Let Sys be a* NegMIF**-system, and p any propositional variable occurring both positively and negatively in Sys. Then p can be eliminated from Sys by the application of the SQEMA-transformation rules. Moreover, the system obtained after the elimination of p will again be a* NegMIF**-system.*

Proof. Let Sys and p satisfy the conditions of the lemma. We will separate out all positive occurrences of p to prepare for the application of the Ackermann-rule. The only positive occurrences of p are heads of (possibly trivial) box-formulae in the consequents of the sequents of the system. Let $\mathbf{i}_i \Rightarrow \alpha$ be such a sequent, where α contains positive occurrences of p. Exhaustive application of the $\wedge$-rule and the $\Diamond$-rule splits $\mathbf{i}_i \Rightarrow \alpha$ into sequents of the forms $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$, $\mathbf{l} \Rightarrow \mathsf{Box}$ and $\mathbf{m} \Rightarrow \mathsf{Neg}$, with $\mathbf{j}, \mathbf{k}, \mathbf{l}, \mathbf{m}$ nominals, Box a box-formula, and Neg a negative formula. As all box-formulae have been left intact, the dependency digraph is unchanged. Hence, after the application of these rules we still have a NegMIF^*-system.

Now it remains to separate the positive occurrences of p out of the equations of the form $\mathbf{l} \Rightarrow \mathsf{Box}$, where Box is of the form $\Box_1(A_1 \vee \Box_2(\ldots \Box_n(A_n \vee p)\ldots))$ where each A_i is a negative formula and each $\Box_i$ a finite, possibly empty, sequence of $\Box$'s. Successive alternative applications of the Left-shift $\vee$-rule and Left-shift $\Box$-rule transforms the sequent

$$\mathbf{l} \Rightarrow A_0 \vee \Box_1(A_1 \vee \ldots \Box_n(A_n \vee p)\ldots))$$

into

$$\neg A_n \wedge \Diamond_n^{-1}(\ldots \Diamond_1^{-1}(\mathbf{l} \wedge \neg A_0)\ldots) \Rightarrow p.$$

The antecedent in the above sequent is a positive formula, not containing p, because the dependency graph is loopless. Hence all positive occurrences of p in the system now occur as the consequents in sequents of the form $\mathsf{Pos} \Rightarrow p$, where Pos is a positive formula not containing p. Let ρ be the disjunction of all the antecedents of the sequents of the form $\mathsf{Pos} \rightarrow p$. Then, by applying the Ackermann-rule, all equations of this form are deleted and p is eliminated by substitution of the positive formula ρ for every negative occurrence of p. Thus, the resulting system does not contain p, all antecedents of equations are nominals, and all consequents of equations are built up from negative formulae and box-formulae, by using only conjunctions and diamonds. Moreover, no positive occurrence of a propositional variable occurs in the scope of an inverse diamond or box.

To show that the resulting system is again NegMIF^*, it remains to show that the dependency digraph is acyclic. We will do so by showing that whenever a new arc (q, u) was introduced by the application of the Ackermann-rule, there was already a directed path from vertex q to vertex u in the digraph *before* the substitution. Indeed, the only way (q, u) could have been introduced was by the substitution of ρ for an inessential occurrence of p in some box-formula with u as head. But then q must occur in ρ, hence, by the construction of ρ, it must have occurred inessentially in some box-formula headed by p. But then (q, p) and (p, u) were arcs in the dependency digraph before the application of the Ackermann-rule, giving the desired path. Thus, the application of the Ackermann-rule cannot introduce cycles in a previously acyclic dependency graph, which completes the argument. QED

Theorem 2.6.8 *SQEMA succeeds on all conjunctions of monadic inductive formulae.*

Proof. An easy inductive argument, noting that the initial SQEMA-systems for every conjunction of monadic inductive formulae are NegMIF^*-systems, and using lemma 2.6.7. QED

Corollary 2.6.9 (Sahlqvist theorem for inductive formulae) *All monadic inductive formulae are elementary and canonical.*

2.7 Computing pure equivalents with SQEMA

Readers might have remarked upon the strangeness of the translation prescribed by step Postprocessing.3 of the SQEMA-algorithm, viz. $\forall\overline{y}\exists x\mathrm{ST}(\neg\mathsf{pure}(\varphi), x)$. After all, why don't we just take the ordinary standard translation of $\neg\mathsf{pure}(\varphi)$? The reason is of course to be found in SQEMA's simulation of local conditions with global statements. This is essential if we are to be able to 'change perspective', as happens in the application of the $\Diamond$ and $\Box$-rules. A plain application of the ordinary standard translation function fails to take this simulation into account, as the following examples illustrate:

Example 2.7.1 In example 2.3.1 we ran SQEMA on the Geach formula $\varphi = \Diamond\Box p \to \Box\Diamond p$, and obtained $\mathsf{pure}(\varphi) = (\neg\mathbf{i}\vee\Diamond\mathbf{j})\wedge(\neg\mathbf{i}\vee\Diamond\Box\Box^{-1}\neg\mathbf{j})$. Hence $\neg\mathsf{pure}(\varphi) = (\mathbf{i}\wedge\Box\neg\mathbf{j})\vee(\mathbf{i}\wedge\Box\Diamond\Diamond^{-1}\mathbf{j})$. Far from defining the Church-Rosser property, this pure formula is locally valid only on those frames consisting of a single (reflexive or irreflexive) point. It is only modulo the interpretation of $\mathbf{i}$ as the current state, and the concomitant special translation, that this formula defines the Church-Rosser property. $\lhd$

Example 2.7.2 The formula $\varphi = \Box(\Box p \to p)$ is valid at a point in a frame iff it has only reflexive successors. If we run SQEMA on this input, we obtain

$$\neg\mathsf{pure}(\varphi) = (\mathbf{i}\wedge\Box\neg\mathbf{j})\vee(\mathbf{j}\wedge\Diamond^{-1}\mathbf{j}).$$

Once again, if we impose no constraints on the interpretation of nominals this formula is locally frame equivalent to $\bot$. Even if we insist that $\mathbf{i}$ be interpreted as the current state, this formula is still falsifiable at points validating $\varphi = \Box(\Box p \to p)$. But let us see how our idiosyncratic translation gets us out of this mess:

$$\begin{aligned} &\forall y_{\mathbf{j}}\exists x\mathrm{ST}((\mathbf{i}\wedge\Box\neg\mathbf{j})\vee(\mathbf{j}\wedge\Diamond^{-1}\mathbf{j}), x)\\ =&\forall y_{\mathbf{j}}\exists x((x = y_{\mathbf{i}}\wedge\forall z_1(Rxz_1 \to z_1 \neq y_{\mathbf{j}})\vee(x = y_{\mathbf{j}}\wedge\exists z_2(Rz_2x\wedge z_2 = y_{\mathbf{j}})) \end{aligned}$$

This formula is a local first-order correspondent of the input formula. Indeed, simplifying it a bit, without any assumptions on interpretations (of variables), we see that it defines the correct local frame condition:

$$\begin{aligned} &\forall y_{\mathbf{j}}(\forall z_1(Ry_{\mathbf{i}}z_1 \to z_1 \neq y_{\mathbf{j}})\vee(\exists z_2(Rz_2y_{\mathbf{j}}\wedge z_2 = y_{\mathbf{j}})) \\ \equiv&\forall y_{\mathbf{j}}(Ry_{\mathbf{i}}y_{\mathbf{j}} \to Ry_{\mathbf{j}}y_{\mathbf{j}}) \end{aligned}$$

$\lhd$

As will be seen, any formula $\varphi\in\mathcal{L}$ reducible by SQEMA is locally equivalent on frames to a pure $\mathcal{L}^n_r$-formula but, as the above examples illustrate, $\neg\mathsf{pure}(\varphi)$ need evidently not be that pure formula. Via the translation $\forall\overline{y}\exists x\mathrm{ST}(\neg\mathsf{pure}(\varphi), x)$ we *do* establish a local correspondence between φ and a first-order formula, but we would also like to use SQEMA to establish correspondences with pure hybrid formulae. This is what we pursue in the remainder of this section. We will call a pure $\mathcal{L}^n_r$-formula α a *pure local frame equivalent* of a $\mathcal{L}$-formula φ, if α and φ are locally equivalent on frames.

The following lemmas are used in justifying the correctness of the procedure that will be proposed further for computing pure local frame equivalents.

Lemma 2.7.3 *Let φ be a $\mathcal{L}_r^n$-formula and $\mathbf{j}$ a nominal, possibly occurring in φ. For any pointed frame $(\mathfrak{F}, w)$, the formula $\mathbf{j} \to \varphi$ is globally $[\mathbf{j} := w]$-satisfiable on $\mathfrak{F}$ iff $(\mathfrak{F}, w) \not\Vdash \mathbf{j} \to \neg\varphi$.*

Lemma 2.7.4 *Any system obtained during the (successful or unsuccessful) execution of SQEMA on any $\mathcal{L}$-input formula, can be transformed, via the application of SQEMA transformation rules, into one in which the antecedent of each sequent is either $\bot$ or a nominal.*

Proof. By an unproblematic induction on the application of transformation rules, we can establish the following:

CLAIM 1 During the entire (successful or unsuccessful) execution of SQEMA on any $\mathcal{L}$-input formula, every sequent $\alpha \Rightarrow \beta$ in every system obtained is such that, after re-normalization, either

1. the antecedent α contains an occurrence either of $\bot$ or of a nominal, which is in the scope of at most $\Diamond^{-1}$ and $\wedge$, or
2. the consequent β contains an occurrence either of $\top$ or of a negated nominal, which is in the scope of at most $\Box$ and $\vee$.

Now, any sequent satisfying the first item of claim 1 can be transformed into the desired form through the application of the right-shift $\vee$ and inverse $\Diamond$-rules. Similarly, a sequent satisfying the second item can be transformed into the desired from through the application of the left-shift $\vee$ and $\Box$-rules, as well as the right-shift $\vee$-rule. QED

We now define a procedure that may be added on to SQEMA to produce the desired pure equivalents:

Procedure Pure Equivalent($\{\mathbf{Sys}_1, \ldots, \mathbf{Sys}_n\}$) This procedure receives a set of pure systems as input, from which it computes and returns a pure $\mathcal{L}_r^n$-formula.

Pure Equivalent.1: Call Unify(Sys_i) for each system Sys_i in $\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$.

Pure Equivalent.2: Form each $1 \leq i \leq n$, the system Unify(Sys_i) consists of a single sequent of the form $\mathbf{i} \Rightarrow \gamma_i$. Form the formula $(\mathbf{i} \to \neg\gamma_1) \wedge \cdots (\mathbf{i} \to \neg\gamma_n)$. Call this formula pure.equiv($\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$).

Pure Equivalent.3: Return pure.equiv($\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$).

Sub-procedure Unify(Sys) This procedure receives a system Sys as input and returns a system consisting of a single sequent of the form $\mathbf{i} \Rightarrow \gamma$.

Unify.1: By applying SQEMA transformation rules, transform Sys into a system in which the antecedent of each sequent $\alpha \Rightarrow \beta$ is either a nominal or $\bot$. (By lemma 2.7.4 this is always possible.)

Unify.2: Remove every sequent with antecedent $\bot$.

Unify.3: The diamond-link sequents in the system define a tree with the occurring nominals as vertices, rooted at the reserved nominal $\mathbf{i}$. Starting from the leaves of this tree (i.e. nominals $\mathbf{k}$ such that there is no diamond-link sequent $\mathbf{k} \Rightarrow \Diamond\mathbf{l}$ in the system), remove all sequents $\mathbf{k} \Rightarrow \beta_1, \ldots \mathbf{k} \Rightarrow \beta_m$ with $\mathbf{k}$ as antecedent and replace the diamond-link sequent $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$ (i.e. the diamond-link sequent with $\mathbf{k}$ in the consequent) with $\mathbf{j} \Rightarrow \Diamond(\mathbf{k} \wedge \beta_1 \wedge \cdots \beta_n)$. Now $\mathbf{j}$ is a leaf in the tree so 'trimmed'. Proceed in this way down the tree towards the root until the only sequents remaining in the system are of the form $\mathbf{i} \Rightarrow \gamma_i$. Replace these with a single sequent $\mathbf{i} \Rightarrow \bigwedge \gamma_i$. Return this system.

By replacing procedure Postprocessing in SQEMA with procedure Pure Equivalent we obtain an algorithm that returns a pure $\mathcal{L}_r^n$-formula rather than a first-order formula. To see that this pure formula will be locally frame equivalent to the input formula we note the following:

1. By proposition 2.4.5 all SQEMA-transformation rules maintain transformation equivalence. Hence step Unify.1 also preserves this equivalence.

2. Step Unify.2 is based on the propositional equivalence of $\bot \to \beta$ and $\top$ and the fact that systems are interpreted conjunctively.

3. The transformations in step Unify.2 clearly preserve global equivalence on models.

4. For any input formula $\varphi \in \mathcal{L}$ and pointed frame $(\mathfrak{F}, w)$, it follows that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathbf{i} \to \neg\varphi$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$ iff $(\mathsf{Unify}(\mathsf{Sys}_1) \vee \cdots \vee \mathsf{Unify}(\mathsf{Sys}_n))$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$, where $\mathsf{Sys}_1, \ldots \mathsf{Sys}_n$ are the systems obtained at the end of phase 2 of a successful execution of SQEMA on φ. Hence $(\mathfrak{F}, w) \Vdash \varphi$ iff $((\mathbf{i} \to \gamma_1) \vee \cdots \vee (\mathbf{i} \to \gamma_n))$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$, with $\gamma_1, \ldots, \gamma_n$ as in Pure Equivalent.2. Hence, by lemma 2.7.3, $(\mathfrak{F}, w) \Vdash \varphi$ iff $(\mathfrak{F}, w) \Vdash ((\mathbf{i} \to \neg\gamma_1) \wedge \cdots \wedge (\mathbf{i} \to \neg\gamma_n))$.

We illustrate this procedure for obtaining pure equivalents with a few examples:

Example 2.7.5 Consider again the Geach formula $\Diamond\Box p \to \Box\Diamond p$ of examples 2.3.1 and 2.7.1. The pure system obtained was

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{i} \Rightarrow \Diamond\Box\Box^{-1}\neg\mathbf{j} \end{array} \right. .$$

Unify changes this into the system

$$\| \ \mathbf{i} \Rightarrow \Diamond\mathbf{j} \wedge \Diamond\Box\Box^{-1}\neg\mathbf{j} \ ,$$

and the pure formula returned is $\mathbf{i} \to \Box\neg\mathbf{j} \vee \Box\Diamond\Diamond^{-1}\mathbf{j}$ which locally defines the Church-Rosser property. ◁

Example 2.7.6 Let us return to the formula $\Box(\Box p \to p)$ of example 2.7.2. The final, pure system obtained when SQEMA is executed on it is

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{j} \Rightarrow \Box^{-1}\neg\mathbf{j} \end{array} \right. .$$

Unify transforms this into the system

$$\| \ \mathbf{i} \Rightarrow \Diamond(\mathbf{j} \wedge \Box^{-1}\neg\mathbf{j}) \ ,$$

and the pure equivalent returned is $\mathbf{i} \rightarrow \Box(\neg\mathbf{j} \vee \Diamond^{-1}\mathbf{j})$ which, as expected, is valid at exactly those points in frames that have only reflexive successors. $\lhd$

Theorem 2.7.7 *SQEMA together with* Pure Equivalent *computes pure local frame equivalents for all Sahlqvist, Sahlqvist–van Benthem, and monadic inductive formulae.*

2.8 SQEMA and van Benthem-formulae

Recall that the van Benthem formulae are those modal formulae that follow locally on frames from L_0 substitution instances of formulae equivalent to their standard translations (definition 1.3.22). In this section we show that SQEMA does not take us beyond that class. The easiest way to see this is via the alternative characterization of the van Benthem-formulae in terms of e-persistence (proposition 1.3.25). We will show that all SQEMA-formulae are e-persistent. With this aim, we adapt the notion of transformation equivalence to elementary frames to obtain:

Definition 2.8.1 Formulae $\varphi, \psi \in \mathcal{L}_r^n$ are *e-transformation equivalent* if, for every model $\mathcal{M} = (\mathfrak{g}, V)$ based on an elementary general frame $\mathfrak{g}$ such that $\mathcal{M} \models \varphi$, there exists a $(\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi), \mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi))$-related model based on $\mathfrak{g}$ such that $\mathcal{M}' \models \psi$, and vice versa.

The following proposition, an analog of proposition 2.4.5, is easy to prove.

Proposition 2.8.2 *If* Sys' *is a SQEMA-system obtained from a system* Sys *by application of SQEMA-transformation rules, then* $\mathsf{Form}(\mathit{Sys})$ *and* $\mathsf{Form}(\mathit{Sys}')$ *are e-transformation equivalent.*

Theorem 2.8.3 *All SQEMA-formulae are locally e-persistent, and hence are van Benthem-formulae.*

Proof. Let $\mathfrak{g} = (\mathfrak{F}, \mathbb{W})$ be an elementary general frame (for the basic modal language $\mathcal{L}$). We note that, apart from the booleans and $\Diamond$, the algebra $\mathbb{W}$ is also closed under $\Diamond^{-1}$ and $\Box^{-1}$. Furthermore, all singletons are admissible. Hence all $\mathcal{L}_r^n$-formulae can be properly interpreted in $\mathfrak{g}$ — no need to consider augmented models here. Moreover, it follows that all pure $\mathcal{L}_r^n$-formulae are e-persistent.

Now suppose $\psi \in \mathcal{L}$ is a SQEMA-formula, and that $\mathsf{pure}(\varphi)$ is the pure $\mathcal{L}_r^n$-formulae obtained in step Postprocessing.2. Now $(\mathfrak{g}, w) \Vdash \varphi$ iff $\mathbf{i} \rightarrow \varphi$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{g}$ iff, by proposition 2.8.2, $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable of $\mathfrak{g}$. By the persistence of pure formulae with respect to elementary frames, the latter is the case iff $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable of $\mathfrak{F}$ which, by proposition 2.4.5, is the case iff $(\mathfrak{F}, w) \Vdash \varphi$. Hence, we have shown that φ is locally e-persistent. QED

This theorem sheds a sharper light on SQEMA's inability to reduce the formula $(\Box p \rightarrow \Box\Box p) \wedge \Box(\Box p \rightarrow \Box\Box p) \wedge (\Box\Diamond p \rightarrow \Diamond\Box p)$ as illustrated in example 2.3.4. According to proposition 1.3.27 this formula is not a van Benthem formula, and hence its reduction will require and algorithm transcending this class. This theorem also suggests that no algorithm based on equivalence preserving transformations (to be precise, transformations preserving logical equivalence between the standard first-order translations of formulae) and substitution (i.e. Ackermann's lemma) will succeed in computing a first-order correspondent for this formula.

Or to be more explicit, suppose that the semantic meaning of the system or formula obtained during any stage of the execution of our hypothetical algorithm can be represented as a monadic existential second-order formula $\exists\overline{P}\varphi$, with φ first-order. If (1) the transformations (apart from the substitution-rule, e.g. Ackermann-rule) performed by the algorithm can be represented by replacing φ with equivalent (in the first-order sense) formulae, and (2) the action of the substitution-rule through which a second-order variables Q is eliminated can be captured by substituting in $\exists Q\exists\overline{P}\varphi$ all occurrences of Q with a first-order formula, then the algorithm maintains equivalence over elementary general frames. Hence, if the algorithm succeeds in reducing an input formula, which we can represent as a second order-formula $\forall\overline{P}\varphi$, to a first-order formula ψ, then $\forall\overline{P}\varphi$ and ψ are equivalent over elementary general frames, and since first-order formulae are e-persistent it follows that $\forall\overline{P}\varphi$ must also be e-persistent. Hence all formulae on which such an algorithm succeeds will be van Benthem–formulae.

Chapter 3

The DLS-Algorithm

As discussed in subsection 1.4.1, the elementarity problem for modal formulae may be seen as a second-order quantifier elimination problem. Particularly, the elementarity question for a modal formula φ is equivalent to the question whether $\forall\overline{P}\mathrm{ST}(\varphi, x)$ is logically equivalent to a formula not containing second-order quantification.

Second-order quantifier elimination is an important problem with a rich diversity of applications (see e.g. [GSS06]). In this chapter we take a closer look at the second-order quantifier elimination algorithm DLS, due to Doherty, Łukaszewicz and Szalas ([DŁS97]). This algorithm has already been introduced briefly in subsection 1.4.3. DLS and SQEMA are related, since both are based on Ackermann's lemma.

Our approach in this chapter can be seen as taking the form of a comparison between instances of the algoritmic and syntactic approaches to specifying classes of elementary (and canonical modal) formulae, as outlined in sections 1.3 and 1.4, respectively. On the one hand we apply the methods of the syntactic approach to the class of formulae defined by DLS, i.e. the class of formulae which it succeeds in reducing, by delineating it partially in terms of its syntactic characteristics. On the other hand we investigate the performance of DLS on two well known syntactically specified classes of modal formulae, namely the Sahlqvist and inductive formulae.

The chapter, which is based on [Con06], is structured as follows: Section 1 provides a characterization of the formulae (of certain syntactic shapes) that are deskolemizable via specified syntactic manipulations. Section 2 recalls and analyses the details of the DLS algorithm. Some examples are also provided. In section 3 we obtain a partial syntactic characterization (in terms of forbidden quantifier-connective patterns) of the second-order formulae in one predicate variable which DLS successfully reduces to first-order formulae. Sufficient conditions for DLS's success on formulae containing multiple existential second-order quantifiers are next provided. Next, in section 4, some consequences of these results for modal correspondence theory are considered. It is shown that DLS succeeds in computing the first-order frame correspondents of all Sahlqvist and inductive formulae. It is proven that all modal formulae in one propositional variable for which DLS computes first-order equivalents are canonical. We conclude in section 5 by mentioning some further directions and conjectures.

3.1 Deskolemization

As will be seen below, the success of the DLS algorithm may depend crucially on the ability to remove Skolem function, i.e., on the ability to deskolemize. In the broadest sense deskolemization[1] may be regarded as the problem of eliminating existentially quantified function variables from second-order formulae whilst maintaining equivalence. As the term suggests, the function variables to be eliminated are typically introduced through Skolemization, i.e. the application of the equivalence

$$\forall\overline{x}\exists y A(\overline{x}, y) \equiv \exists f \forall\overline{x} A(\overline{x}, y)[f(\overline{x})/y] \tag{3.1}$$

from left to right in order to eliminate existential quantifiers. Note that the equivalence assumes the axiom of choice. Some methods for deskolemizing clause sets are provided in [Eng96] and [McC88]. Here, however, we are interested in deskolemizing a rather particular kind of formulae, viz. those produced by DLS after the application of Ackermann's lemma in phase 3 (see section 3.3). With this in mind we consider a slightly more general situation and introduce the following definitions.

We define the following transformations on predicate formulae, based on well known equivalences. The arrows indicate in which directions the transformations are applied. (Certain asymmetries in the list, e.g. the separation of clause (Q2) and (Q3) unlike clause (Q1), are motivated by the combinations in which these transformations will be applied, which in turn is dictated by the specification of the DLS-algorithm.)

(Q1) $\mathsf{Q}x(A * B(x)) \Longrightarrow A * \mathsf{Q}xB(x)$, and $\mathsf{Q}x(B(x) * A) \Longrightarrow \mathsf{Q}xB(x) * A$, for $\mathsf{Q} \in \{\exists, \forall\}$, $* \in \{\wedge, \vee\}$ and where A contains no free occurrence of the variable x.

(Q2) $A * \exists xB(x) \Longrightarrow \exists x(A * B(x))$ and $\exists xB(x) * A \Longrightarrow \exists x(B(x) * A)$, for $* \in \{\wedge, \vee\}$, and where A contains no free occurrence of the variable x.

(Q3) $A * \forall xB(x) \Longrightarrow \forall x(A * B(x))$ and $\forall xB(x) * A \Longrightarrow \forall x(B(x) * A)$, for $* \in \{\wedge, \vee\}$, and where A contains no free occurrence of the variable x.

(Q4) $\forall x(A \wedge B) \Longrightarrow \forall xA \wedge \forall xB$, where A and B are any formulae.

(Q5) $\forall xA \wedge \forall xB \Longrightarrow \forall x(A \wedge B)$, where A and B are any formulae.

(Q6) $\exists x(A \vee B) \Longrightarrow \exists xA \vee \exists xB$, where A and B are any formulae.

(Q7) $\exists xA \vee \exists xB \Longrightarrow \exists x(A \vee B)$, where A and B are any formulae.

(Q8) $\exists x\exists yA \Longrightarrow \exists y\exists xA$ and $\forall x\forall yA \Longrightarrow \forall y\forall xA$ for any formula A.

A first-order formula is in *negation normal form* if it contains no occurrences of '$\rightarrow$' and '$\leftrightarrow$' and all negation signs occur only directly in front of atomic formulae. A first-order formula is said to be *clean* if no variable occurs both bound and free, and no two quantifier occurrences

[1] The term *unskolemization* is probably more widely used. However, because of the possible ambiguity of certain forms like '*unskolemizable*', we prefer *deskolemization* and *deskolemizable*.

bind the same variable. Clearly every first-order formula may be equivalently rewritten as a clean formula by a suitable renaming of bound variables. The scope of an occurrence of a quantifier $\mathsf{Q} \in \{\forall, \exists\}$ is *minimal* (respectively, *strongly minimal*) in a first-order formula if that occurrence can not be moved to the right (i.e. if its scope cannot be made smaller) by the application of (Q1) (respectively, of (Q1) and (Q4)).

Definition 3.1.1 A first-order formula A is (strongly) *standardized* with respect to a predicate symbol P if it is clean, in negation normal form, and the scope of all quantifiers in the scope of which P occurs, is (strongly) minimal.

Definition 3.1.2 A formula is in *deskolemizable form with respect to* $f_1, \dots, f_n$ if it is clean and has the form

$$\exists f_1 \dots \exists f_n \forall \overline{x} A$$

where for each f_i,

(i) all arguments of each occurrence of f_i are variables among $\overline{x}$,

(ii) each occurrence of f_i is applied to the same vector of variable arguments, i.e to the same variables in the same order, and

(iii) the ordering of the arguments of the f_i's by set inclusion is linear (but not necessarily strict).

Definition 3.1.3 A formula is in *general deskolemizable form with respect to* $f_1, \dots, f_n$, if it is built up from formulae not containing any occurrences of function variables among $f_1, \dots, f_n$ and formulae in deskolemizable form with respect to function variables among $f_1, \dots, f_n$, by applying the boolean connectives and individual quantifiers.

Note that the function variables $f_1, \dots, f_n$ may be eliminated from any formula in deskolemizable form with respect to $f_1, \dots, f_n$, via the application of (3.1) from right to left, slotting in the new existential individual quantifiers into the prefix $\forall \overline{x}$ appropriately so as to get the variable dependencies right. This observation readily extends to yield the following proposition:

Proposition 3.1.4 *The function variables $f_1, \dots, f_n$ can be eliminated from any formula in general deskolemizable form with respect to $f_1, \dots, f_n$, via the application of (3.1).*

Incidentally, every formula satisfying all conditions of definition 3.1.2, except perhaps condition (iii), can be equivalently rewritten as a first-order formula with Henkin-quantifiers ([Wal70]).

Or approach to deskolemization will be to try and transform formulae into general deskolemizable form via equivalence preserving syntactic manipulations.

By *quantifier shifting* we will mean the application of the rules (Q1) to (Q8). By a *top level existential quantifier* in a formula A we mean any occurrence of an existential quantifier in A which is not in the scope of any universal quantifier. Similarly, a *top level universal quantifier* is a universal quantifier not in the scope of any existential quantifier. An occurrence of a conjunction or disjunction which is not in the scope of any universal quantifier is called a *top level conjunction / disjunction*.

Definition 3.1.5 Let A be a formula in negation normal form and P a predicate variable. A conjunction (disjunction) occurrence in A is called:

1. *benign with respect to* P, if it is the main connective of a subformula of the form $C \wedge D$ ($C \vee D$) where at least one of C and D contains no occurrences of P. We will write $\wedge'_P$ ($\vee'_P$) for conjunctions (disjunctions) benign with respect to P, or simply $\wedge'$ ($\vee'$) if P is understood;

2. *malignant with respect to* P if it is the main connective of a subformula of the form $C \wedge D$ ($C \vee D$) where C and D contains occurrences of P of opposite polarity. We will write conjunctions (disjunctions) malignant with respect to P as $\wedge^*_P$ ($\vee^*_P$), or simply as $\wedge^*$ ($\vee^*$) when P is understood;

3. *non-malignant with respect to* P, if it is not malignant with respect to P. Clearly benign connectives are non-malignant, but not conversely. We will write $\wedge^\circ_P$ ($\vee^\circ_P$) for conjunctions (disjunctions) non-malignant with respect to P, or simply $\wedge^\circ$ ($\vee^\circ$) if P is understood;

4. *non-benign with respect to* P, if it is not benign with respect to P. We will not associate a special notation with non-benign conjunctions and disjunctions.

Example 3.1.6 In the formula

$$\forall x(\neg Q(x) \vee' P(x)) \wedge^* \exists u((P(u) \wedge' Q(u)) \wedge^\circ \forall y(\neg R(u,y) \vee' \exists z(R(y,z) \wedge' Q(z)))),$$

we have indicated for all conjunctions and disjunctions whether they are benign, malignant or non-malignant with respect to Q. $\lhd$

The following lemma is easy to prove:

Lemma 3.1.7 *Let A be a formula, P a predicate symbol, and suppose that A' is obtained from A by (i) distributions of $\wedge$ over $\vee$, and/or (ii) distribution of $\vee$ over $\wedge$, and/or (iii) the application of the associativity laws $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$ and $((A \vee B) \vee C) \equiv (A \vee (B \vee C))$. Then A contains a conjunction (disjunction) malignant/non-benign with respect to P only if A' contains a conjunction (disjunction) malignant/non-benign with respect to P.*

Definition 3.1.8 A predicate P, is in $\exists\forall$-*scope* in a formula A, if no occurrence of P in A is in the scope of

(i) an existential quantifier which is in the scope of a universal quantifier, or

(ii) a non-benign (w.r.t. P) disjunction which is in the scope of a universal quantifier, or

(iii) a malignant (w.r.t. P) conjunction which is in the scope of a universal quantifier.

Example 3.1.9 In the formula

$$\forall x[\neg Q(x) \vee P(x)] \wedge \exists u[P(u) \wedge \forall y(\neg R(u,y) \vee \exists z(R(y,z) \wedge Q(z)))],$$

which is clean and in negation normal form, P is in $\exists\forall$-scope, while Q is not in $\exists\forall$-scope. $\lhd$

Lemma 3.1.10 *Let $\exists f_1 \ldots \exists f_n \forall \overline{x} A(\overline{x}, y)$ be a formula in deskolemizable form with respect to $f_1 \ldots f_n$, in which the variable y is free and does not occur as an argument of any $f_1 \ldots f_n$. Let $B(P)$ be a formula standardized with respect to the unary predicate P, positive (negative) in P, in which the variables $\overline{x}, y$ do not occur. Let $B(\forall \overline{x} A/P)$ be the result of substituting all occurrences of P ($\neg P$) in B with $\forall \overline{x} A(\overline{x}, y)$, the actual argument of each occurrence of P every time being substituted for y in A. Then the formula*

$$\exists f_1 \ldots \exists f_n B(\forall \overline{x} A/P)$$

can be transformed into general deskolemizable form by quantifier shifting and the distribution of conjunction over disjunction, if and only of P is in $\exists\forall$-scope in $B(P)$.

Proof. Suppose P is in $\exists\forall$-scope in $B(P)$. Note that, since B is positive (negative) in P, it contains no occurrences of $\wedge$ or $\vee$ which are malignant with respect to P. We show how $\exists f_1 \ldots \exists f_n B(\forall \overline{x} A/P)$ may be brought into general deskolemizable form by means of quantifier shifting and the distribution of conjunctions over disjunctions. First pull out all top level existential quantifiers (by applying (Q2)), and then distribute top-level conjunctions over top-level disjunctions as much as possible, to obtain

$$\exists \overline{z} \exists f_1 \ldots \exists f_n B'(\forall \overline{x} A/P). \tag{3.2}$$

Note that, in $B'(P)$, P is still in $\exists\forall$-scope, and that no non-benign disjunction occurs in the scope a conjunction. Now, in (3.2), the prefix $\exists f_1 \ldots \exists f_n$ may be shifted to the right across all occurrences of $\vee$ and $\wedge'_P$, using (Q6) and (Q1), respectively. On the other hand, occurrences of the prefix $\forall \overline{x}$ may be shifted to the left across all occurrences of $\wedge$ and $\forall$ using (Q3), (Q5) and (Q8), as well as occurrences of $\vee'_P$, using (Q3). This may be done until each occurrence of $\forall \overline{x}$ stands directly to the right of an occurrence of $\exists f_1 \ldots \exists f_n$, and vice versa. Moreover, within the scope of each occurrence of $\exists f_1 \ldots \exists f_n$, conditions (i) to (iii) of definition 3.1.2 will be satisfied with respect to $f_1 \ldots f_n$, since the process described does not affect the arguments of occurrences of the $f_1 \ldots f_n$. Hence, the resulting formula will be in general deskolemizable form with respect to $f_1 \ldots f_n$.

Conversely, if P is out of $\exists\forall$-scope in $B(P)$, it means that condition (i) or (ii) of definition 3.1.8 is violated in $B(P)$ — condition (iii) will not be violated, since $B(P)$ is positive (negative) in P. Note that these situations may not be removed by distribution and/or quantifier shifting, as, by assumption, all quantifiers in $B(P)$ with P in their scope already have minimal scope. Hence quantifier shifting and distribution of $\wedge$ over $\vee$ will fail here to bring the formula into general deskolemizable form, as it will fail to make each occurrence of $\exists f_1 \ldots \exists f_n$ stand directly to the left of some occurrence of $\forall \overline{x}$, or to make every occurrence of $\forall \overline{x}$ stand directly to the right of some occurrence of $\exists f_1 \ldots \exists f_n$. QED

Example 3.1.11 Consider the formula $\exists f \forall w A(w, y)$, where A is $(y \neq f(w) \vee \neg Rxw)$, which is in deskolemizable form. If B is the formula $\forall u(\neg Rxu \vee \exists v(Ruv \wedge \neg P(v)))$ or $\forall u \forall v(\neg Rxu \vee \neg P(u) \vee (Ruv \wedge \neg P(v)))$, then the formula $\exists f B(\forall w A/P)$ cannot be transformed into deskolemizable form, since, in both cases, P is not in $\exists\forall$-scope in B. However, if B is $\forall u(\neg Rxu \vee \forall v(\neg Ruv \vee \neg P(v)))$, then $\exists f B(\forall w A/P)$ can be transformed into deskolemizable form. $\lhd$

3.2 The DLS algorithm

In this section we recount the full details of the DLS algorithm as presented in [DŁS97], interpolating remarks, relevant to the purposes of this chapter, into the exposition. We spell out the assumptions we make regarding certain aspects of the algorithm, left unspecified in [DŁS97], but crucial to some results in the subsequent sections. For some fully worked-out examples of the (successful and unsuccessful) execution of DLS on various input formulae, the reader is referred to [DŁS97] and, for modal examples, to [Sza93] and [Sza02].

DLS is centered around the equivalences given in Ackermann's Lemma ([Ack35]), already introduced in chapter 2, but restated here for easy reference:

Lemma 3.2.1 (Ackermann's Lemma) *Let $A(\overline{z}, x)$ be a formula not containing P. Then, if, $B(P)$ is negative in P, the equivalence*

$$\exists P \forall x((\neg A(\overline{z}, x) \vee P(x)) \wedge B(P)) \equiv B[A(\overline{z}, x)/P] \tag{3.3}$$

holds, with $B[A(\overline{z}, x)/P]$ the formula obtained by substituting $A(\overline{z}, x)$ for all occurrences P in B, the actual argument of each occurrence of P being substituted for x in $A(\overline{z}, x)$ every time. If $B(P)$ is positive in P, a similar equivalence holds:

$$\exists P \forall x((\neg P(x) \vee A(\overline{z}, x)) \wedge B(P)) \equiv B[A(\overline{z}, x)/P]. \tag{3.4}$$

The algorithm takes as input a formula $\exists P A$ where no second order quantification occurs in A. To eliminate multiple existentially quantified predicate variables, the algorithm may be iterated, and to eliminate universally quantified predicate variables, the negation of the formula can be considered. For simplicity we assume that all predicate variables are unary, which is sufficient for modal applications. The execution of algorithm consists of the following four phases. (Examples 3.2.9 and 3.2.10 at the end of this section illustrate these phases of algorithm.)

3.2.1 Phase 1: preprocessing

The purpose of this phase is to separate positive and negative occurrences of P by transforming the input formula $\exists P A$ into the form

$$\exists \overline{x} \exists P[(A_1(P) \wedge B_1(P)) \vee \cdots \vee (A_n(P) \wedge B_n(P))], \tag{3.5}$$

where each $A_i(P)$ (respectively, $B_i(P)$) is positive (respectively, negative) in P. If this cannot be achieved, the algorithm reports failure and terminates.

1. Make the formula clean by renaming bound variables, and transform it into negation normal form.

2. Universal quantifiers are moved to the right as far as possible by applying (Q1), and then existential quantifiers are moved to the left as far as possible using (Q2).

3. Move existential quantifiers in the scope of universal quantifiers to the right, using (Q1).

4. Steps (2) and (3) are repeated as long as new existential quantifiers can be moved into the prefix.

   Remark 3.2.2 Note that, if A is standardized with respect to P, then the only effect of steps (2) to (4) on quantifiers with P in their scope, will be to pull out into the prefix the top level $\exists$'s among them. For strongly standardized A, this will even be the case if, in step 2, the application of (Q4) is moreover allowed.

5. In the matrix of the formula obtained thus far, distribute conjunctions over disjunctions that contain both positive and negative occurrences of P, i.e. replace $A \wedge (B \vee C)$ with $(A \wedge B) \vee (A \wedge C)$ whenever $(B \vee C)$ contains both positive and negative occurrences of P.

If after these 5 steps the desired separated form (3.5) has not been obtained, the algorithm reports failure. Otherwise, transform the obtained formula into its equivalent

$$\exists \overline{x}[\exists P(A_1(P) \wedge B_1(P)) \vee \cdots \vee \exists P(A_n(P) \wedge B_n(P))]. \tag{3.6}$$

Phase 2 then proceed separately on each main disjunct $\exists P(A_i(P) \wedge B_i(P))$.

Remark 3.2.3 We assume that the algorithm has some mechanism built in to deal with the associativity and commutativity of $\wedge$ and $\vee$. This is reasonable, for it is clear that without such a mechanism phase 1 will fail to solve such clearly solvable input as $\exists P \exists x \exists y \exists z((P(x) \wedge Q(x) \wedge \neg P(y)) \wedge (P(z)))$. Moreover this mechanism should be optimizable to minimize either negative or positive conjuncts. For example, in the above formula, should $Q(x)$ be included in the conjunct negative or positive with respect to P? Minimization with respect to positive conjuncts would yield a formula $\exists P \exists x \exists y \exists z((P(x) \wedge P(z)) \wedge (Q(x) \wedge \neg P(y)))$, while minimization with respect to negative conjuncts will give $\exists P \exists x \exists y \exists z((P(x) \wedge Q(x) \wedge P(z)) \wedge \neg P(y))$.

3.2.2 Phase 2: preparation for Ackermann's lemma

For this phase to be reached, it is necessary that the input formula has been successfully transformed into the form (3.6). This phase transforms a formula of the form $\exists P(A \wedge B)$, with A and B respectively positive and negative in P, into one of the two forms, (3.3) or (3.4), suitable for the application of Ackermann's lemma. Both forms are always obtainable. However, it is possible that one form may lead to failure of the deskolemization in phase 3, while the other does not. Accordingly both forms are obtained, in order to increase the chances of success. We outline the transformation procedure used to obtain the first form — the other is symmetric.

1. Transform A into prenex conjunctive normal form, using the usual method. We obtain a formula of the form

$$\mathit{pref}\,[(P(t_{1_1}) \vee \cdots \vee P(t_{m_1}) \vee C_1) \wedge \cdots \wedge (P(t_{1_k}) \vee \cdots \vee P(t_{m_k}) \vee C_k) \wedge D], \tag{3.7}$$

where *pref* is a quantifier prefix and P does not occur in $C_1, \ldots, C_k, D$.

Remark 3.2.4 In preparing for the application of Ackermann's lemma, the goal of phase 2 is to 'extract' the occurrences of P. In order to do this, subformulae not containing occurrences of P need not be transformed in any way. Accordingly, step 1 of this phase may be optimized as follows: rather than obtaining a full prenex conjunctive normal form, pull into the prefix only quantifiers that have P in their scope; then distribute disjunctions over conjunctions which do not occur in the scope of quantifiers other than those in the prefix. Note that the obtained formula (3.7) will be the same, except that the C_i and D need not be quantifier free. Proceeding in this way minimizes the introduction of (existential) quantifiers into the prefix *pref*, and hence the introduction of Skolem functions in step 4, below. In the implementation of DLS (see [Gus96]) similar strategies are used. It is not difficult to construct formulae on which DLS would fail if a full prenex conjunctive form were obtained, but on which it succeeds if the described strategy is followed, and indeed on which the implementation also succeeds.

2. Transform each conjunct of (3.7) of the form $(P(t_{1_i}) \vee \cdots \vee P(t_{m_i}) \vee C_i)$ (i.e. each conjunct with multiple P-disjuncts) equivalently into

$$\exists x_i(\forall y(P(y) \vee x_i \neq y \vee C_i) \wedge (x_i = t_{1_i} \vee \cdots \vee x_i = t_{m_i} \vee C_i)).$$

Move each new existential quantifier $\exists x_i$ into the prefix *pref*, and move each of the conjuncts $(x_i = t_{1_i} \vee \cdots \vee x_i = t_{m_i} \vee C_i)$ into D in (3.7), renaming D to D'.

Remark 3.2.5 The new existential quantifiers being introduced into *pref* will have to be skolemized in step 4 below.

3. Transform each conjunct of (3.7) of the form $P(t_{1_i}) \vee C_i$ (i.e. each conjunct with only one P-disjunct) equivalently into the form $\forall y(P(y) \vee y \neq t_{1_i} \vee C_i)$.

4. Skolemize all existential quantifiers in the prefix of the formula obtained so far. The input to this phase has now been transformed into the form

$$\exists \overline{f} \exists P \mathit{pref}'[\forall y(P(y) \vee x_1 \neq y \vee C_1) \wedge \cdots \wedge \forall y(P(y) \vee x_k \neq y \vee C_k) \wedge E], \tag{3.8}$$

where E is $D' \wedge B$. Note that this may cause some of the x_i to be replaced with Skolem functions, and that Skolem functions may be introduced into the C_i and also into D'.

5. Lastly transform (3.8) into the form

$$\exists \overline{f} \exists P \forall y[P(y) \vee \mathit{pref}'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k)) \wedge \mathit{pref}' E]. \tag{3.9}$$

Remark 3.2.6 We note that the formula $\exists\overline{f}[pref'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))]$ is in deskolemizable form. Indeed, the arguments of all occurrences of the introduced Skolem functions $\overline{f}$ are variables among those bound in $pref'$, and are not changed from those inserted during the skolemization step.

3.2.3 Phase 3: application of Ackermann's lemma

The formula (3.9) obtained in the last step of the previous phase is of the right shape to permit the application of Ackermann's lemma. Accordingly the algorithm proceeds as follows:

1. Apply Ackermann's lemma to (3.9), eliminating P and obtaining the formula

$$\exists\overline{f}pref'E[pref'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))/\neg P]. \tag{3.10}$$

   Remark 3.2.7 Recall that E is the formula $D' \wedge B$. Since the formula was made clean in step 1 of phase 1, no variable bound by a quantifier in $pref'$ occurs in B. Moreover, by the construction of D' it contains no occurrences of P.

2. Deskolemize if possible, by applying the equivalence (3.1). If this is not possible the algorithm reports failure and terminates.

   Remark 3.2.8 The deskolemization step is rather underspecified. It should be clear that the formula obtained in step 1 of this phase will rarely be in a form to which (3.1) is directly applicable. For these reasons we make the assumption that the deskolemization further involves quantifier shifting and distribution $\wedge$ over $\vee$ (as in the proof of lemma 3.1.10) to try and bring the formula into general deskolemizable form, from which the Skolem functions may then be eliminated by the application of (3.1).

3.2.4 Phase 4: simplification

In this phase the following simplifying substitutions are performed on the formula obtain in phase 3: subformulae of the form $\forall x(A(x) \vee x \neq t)$ are replaced by $A(t)$, and subformulae of the form $\forall x((x \neq t_1 \wedge \cdots \wedge x \neq t_n) \vee A(x))$ are replaced with $A(t_1) \wedge \cdots \wedge A(t_n)$.

3.2.5 Examples

Here are two examples of the execution of DLS on the negations of translations of modal formulae.

Example 3.2.9 In example 2.3.1 we saw how SQEMA succeeds in computing a first-order equivalent for the Geach formula, $\Diamond\Box p \to \Box\Diamond p$. This is how DLS deals with the negation of its translation $\exists P(\exists y(Rxy \wedge \forall z(Ryz \to P(z))) \wedge \exists u(Rxu \wedge \forall v(Ruv \to \neg P(v))))$:

$$\begin{aligned} & \exists P(\exists y(Rxy \wedge \forall z(Ryz \to P(z))) \wedge \exists u(Rxu \wedge \forall v(Ruv \to \neg P(v)))) \\ \equiv\ & \exists y \exists u \exists P((Rxy \wedge \forall z(\neg Ryz \vee P(z))) \wedge (Rxu \wedge \forall v(\neg Ruv \vee \neg P(v)))) \end{aligned}$$

Phase 1 completed. Phase 2, step 1:

$$\equiv \exists y \exists u \exists P(\forall z((\neg Ryz \lor P(z)) \land Rxy) \land (Rxu \land \forall v(\neg Ruv \lor \neg P(v))))$$

Phase 2, step 3:

$$\equiv \exists y \exists u \exists P(\forall z(\forall w(P(w) \lor w \neq z \lor \neg Ryz) \land Rxy) \land (Rxu \land \forall v(\neg Ruv \lor \neg P(v))))$$

Phase 2, step 3:

$$\equiv \exists y \exists u \exists P \forall w((P(w) \lor \forall z(w \neq z \lor \neg Ryz)) \land \forall z(Rxy \land (Rxu \land \forall v(\neg Ruv \lor \neg P(v)))))$$

Phase 3, step 1:

$$\equiv \exists y \exists u \forall z(Rxy \land (Rxu \land \forall v(\neg Ruv \lor \neg \forall z(v \neq z \lor \neg Ryz))))$$

After negation this formula simplifies to

$$\forall y \forall u(Rxy \land Rxu \to \exists v(Ruv \land Ryv))$$

as expected. $\lhd$

Example 3.2.10 Consider the negated translation of the formula $\Box(\Box p \leftrightarrow q) \to p$ from example 2.3.3. Phase 1 of DLS begins by making this formula clean and transforming it into negation normal form, yielding

$$\forall z(\neg Rxz \lor ((\neg Q(z) \lor \forall u(\neg Rzu \lor P(u))) \land (Q(z) \lor \exists v(Rzv \land \neg P(v))))) \lor \neg Q(x)$$

By remark 3.2.2, phase 1 of DLS will fail to bring this formula into the desired form (3.5). Hence the algorithm reports failure. $\lhd$

3.3 Characterizing the success of DLS

In this section we attempt to gain a better understanding of the input formulae on which DLS will succeed. Given our assumptions about the deskolemization process (see remark 3.2.8), we are in fact able to give a precise syntactic characterization of the formulae $\exists PA$, with A standardized with respect to P, from which DLS will succeed in eliminating the predicate variable P. However, when it comes to the iterated elimination of multiple predicate variables, the situation is significantly more complicated, and in this case we content ourselves with providing sufficient syntactic conditions for the success of the algorithm.

3.3.1 A necessary and sufficient condition for success

Definition 3.3.1 A predicate variable P is in *good scope* in a formula A in negation normal form if

(i) no conjunction or disjunction malignant with respect to P occurs in the scope of a universal quantifier in A, and

(ii) A contains no subformula of the form $B_1 \wedge B_2$ where P is out of $\exists\forall$-scope in both B_1 and B_2, and B_1 contains a positive (negative) occurrence of P, and B_2 contains a negative (positive) occurrence of P.

A predicate symbol P is in *bad scope* in a formula A, if it is not in good scope in A.

Example 3.3.2 In the formulae

$$\forall x[\exists y(R(x,y) \wedge \neg P(y)) \wedge \forall y(\neg R(x,y) \vee P(y))]$$

and

$$[\forall z \exists y(R(z,y) \wedge P(y))] \wedge [\forall z \exists y(R(z,y) \wedge \neg P(y))]$$

the predicate symbol P is in bad scope. However, it is in good scope in the formulae

$$\exists x[\exists y(R(x,y) \wedge \neg P(y)) \wedge \forall y(\neg R(x,y) \vee P(y))]$$

and

$$[\forall z \forall y(R(z,y) \wedge P(y))] \wedge [\forall z \exists y(R(z,y) \wedge \neg P(y))].$$

P is in good scope in any formula that is positive (negative) in P. $\triangleleft$

Theorem 3.3.3 *Let A be a formula, standardized with respect to P. Then DLS succeeds in eliminating P from $\exists PA$ if and only if P is in good scope in A.*

Proof. We prove a number of subclaims. The first claim is easy to see.

Claim 1 Let C be a formula and P a predicate symbol. Let C' be obtained from C by (i) pulling out top level $\exists$, and/or (ii) distributions of top-level $\wedge$'s over top-level $\vee$'s. Then P is in good scope in C if and only if P is in good scope in C'. Moreover, disregarding top level $\exists$'s, C' is standardized w.r.t. P whenever C is.

Claim 2 If P is in good scope in A, then $\exists PA$ may be transformed into the shape given by (3.5) by pulling out into the prefix all top level $\exists$'s, and by distributing conjunctions over malignant disjunctions. Moreover, P will still be in good scope in the resulting formula.

Proof of Claim If P is in good scope in A, then all $\wedge^*_P$'s and $\vee^*_P$'s are top-level. Hence, after pulling top level $\exists$'s into the prefix and distributing top level conjunctions over malignant disjunctions, the formula will be in the desired shape, modulo associativity and commutativity of $\wedge$ and $\vee$ (see remark 3.2.3). By claim 1, this procedure preserves the good scope of P. ◀

Combining claim 2 with remark 3.2.2, and noting step 5 in Phase 1, it follows that Phase 1 of DLS will succeed on any formula $\exists PA$ with A standardized, whenever P is in good scope in A.

Claim 3 Suppose P is in bad scope in A, and that phase 1 of the algorithm succeeds in transforming $\exists PA$ into the desired shape (3.5). Then P will be in bad scope in at least one of the main disjuncts $\exists P(A_i(P) \wedge B_i(P))$ of formula (3.6).

Proof of Claim We show that the obtained formula (3.6) will contain a main disjunct $\exists P(A_i(P) \wedge B_i(P))$ such that neither in $A_i(P)$ nor in $B_i(P)$ the predicates symbol P is in $\exists\forall$-scope. By claim 1, P will be in bad scope in the matrix of the formula (3.6). It cannot be the case that the first condition of definition 3.3.1 is violated, since then negative and positive occurrences of P could not be separated in the formula as per assumption. Hence it must be the second condition which is violated, i.e. there is a subformula of the form $(B_1 \wedge B_2)$ with a positive (negative) occurrence of P not in $\exists\forall$-scope in B_1 and a negative (positive) occurrence of P not in $\exists\forall$-scope in B_2. Then, since the only malignant conjunctions in our formula are those between the A_i-B_i-pairs, the claim follows. ◀

Claim 4 Given as input to phase 2 a formula $\exists P(A(P) \wedge B(P))$, standardized with respect to P and with $A(P)$ positive and $B(P)$ negative in P, DLS will terminate successfully if and only if P is in good scope in $(A(P) \wedge B(P))$.

Proof of Claim Phase 2 transforms $\exists P(A(P) \wedge B(P))$ into the form (3.9), which step 1 of phase 3 transforms into $\exists \overline{f} \mathit{pref}' E[\mathit{pref}'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))/\neg P]$, where E is $D' \wedge B(P)$ and D' contains no occurrences of P. By remark 3.2.7, this formula can be transformed by quantifier shifting into the form $\exists \overline{f}(\mathit{pref}' D' \wedge B[\mathit{pref}'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))/\neg P])$. By remark 3.2.6, $\exists \overline{f}[\mathit{pref}'((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))]$ is in deskolemizable form. If then follows from lemma 3.1.10 and our assumptions on the deskolemization process (see remark 3.2.8), that the next step of the algorithm will succeed in deskolemizing this formula if and only if P is in $\exists\forall$-scope in B.

Now, suppose P is in bad scope in $(A(P) \wedge B(P))$, and hence P is in $\exists\forall$-scope neither in $A(P)$ nor in $B(P)$. Then skolem functions will be introduced during phase 2 and, by the above, the deskolemization and the algorithm will fail, and this will also be the case if, rather than version (3.3) of Ackermann's lemma, version (3.4) is prepared for and applied.

Conversely, if P is in good scope in $(A(P) \wedge B(P))$, then it is in $\exists\forall$-scope in at least one of $A(P)$ or $B(P)$. Hence the algorithm will succeed, either when version (3.3) Ackerman's lemma is prepared for and applied (as illustrated above), or when version (3.4) is used. ◀

Theorem 3.3.3 now follows by combining claims 2 through 4. QED

Remark 3.3.4 If (Q4) is also used in phase 1 (see remark 3.2.2) theorem 3.3.3 will hold for strongly standardized, rather than standardized, formulae. Note that the theorem does *not* extend to the elimination of multiple predicate variables: both P and Q are in good scope in $\exists P \exists Q \exists x[\forall y(\neg Rxy \vee (P(y) \vee \forall z(\neg Ryz \vee Q(z)))) \wedge \forall u(\neg Rxu \vee \exists v(Ruv \wedge (\neg P(v) \vee \neg Q(v))))]$, but one can easily check that DLS will fail to reduce this formula.

3.3.2 A sufficient condition for success

Definition 3.3.5 A formula A is *restricted* with respect to predicate variables $P_1, \ldots, P_n$ if it is standardized with respect $P_1, \ldots, P_n$, and, in A,

(i) no two positive occurrences of predicate variables among $P_1, \ldots, P_n$ are in the scope of the same universal quantifier, and

(ii) all positive occurrences of $P_1, \dots, P_n$ are in $\exists\forall$-scope.

The next definition is inspired by that of the inductive formulae (definition 4.2.3).

Definition 3.3.6 Let A be a restricted formula with respect to $P_1, \dots, P_n$. The *dependency digraph of A over $P_1, \dots, P_n$* is the digraph $D_A = \langle V_A, E_A \rangle$ with vertex set $V_A = \{P_1, \dots, P_n\}$ and edge set E_A such that $(P_i, P_j) \in E_A$ iff there is a subformula $\forall x C$ of A such that P_i occurs negatively in C and P_j occurs positively in C. The dependency digraph of A is *acyclic* if it contains no directed cycles or loops. A formula A, restricted with respect to $P_1, \dots, P_n$, is *independent* with respect to $P_1, \dots, P_n$ if its dependency digraph over $P_1, \dots, P_n$ is acyclic.

Example 3.3.7 Consider the formulae $\forall x(P(x) \vee \forall z(R(x,z) \wedge Q(z)))$ and $\forall x(P(x) \vee \forall z(R(x,z) \wedge \neg Q(z))) \wedge \forall x(\forall y(\neg R(x,y) \vee Q(y)) \vee \exists z(R(x,z) \wedge \neg P(z)))$. The first is not restricted with respect to P and Q. The second is restricted with respect to P and Q but not independent with respect to these predicate variables. It can be made independent by replacing the subformula $\neg Q(z)$ with $z \neq z$, for instance. $\lhd$

Lemma 3.3.8 *Suppose A is independent with respect to $P_1, \dots, P_n$, and that P_1 is minimal with respect to the ordering induced on $P_1, \dots, P_n$ by the dependency digraph. Then DLS succeeds in eliminating P_1 from $\exists P_1 A$. Moreover, the returned formula will be independent with respect to $P_2, \dots, P_n$.*

Proof. We assume that A has been preprocessed by distributing all $\wedge$'s and $\exists$'s over $\vee$ as much as possible.

CLAIM 1 Stage 1 terminates successfully, returning a formula $\exists \overline{x}[\exists P(A_1(P_1) \wedge B_1(P_1)) \wedge \cdots \wedge \exists P(A_n(P_1) \wedge B_n(P_1))]$ in which each $(A_i(P) \wedge B_i(P))$ is independent with respect to $P_1, \dots, P_n$, and such that in each $A_i(P_1)$ the only occurrences of predicate variables among $P_1, \dots, P_n$ are P_1's. Moreover, no $A_i(P_1)$ has a subformula of the form $C \vee D$ with P_1 occurring both in C and D.

PROOF OF CLAIM We note that P_1 is in good scope in A, and hence that, by theorem 3.3.3, phase 1 succeeds. By the minimality of P_1, no occurrence of a predicate variable among $P_2, \dots, P_n$ occurs together with P_1 in the scope of a universal quantifier. Hence, apart from top level $\exists$'s, the only occurrences of connectives in the scope of which P_1 occurs in A together with other predicate variables among $P_2, \dots, P_n$, are conjunctions and disjunctions. Moreover, among these conjunctions and disjunctions, no disjunction occurs in the scope of a conjunction, due to the preprocessing. It follows that, if we minimize positive conjuncts (see remark 3.2.3), in the formula (3.6) no conjunct $A_i(P_1)$ will contain any occurrence of predicate symbol among $P_2, \dots, P_n$, nor will it contain a subformula of the form $C \vee D$ with P_1 occurring both in C and D. Moreover, each main disjunct $\exists P_1(A_i(P_1) \wedge B_i(P_1))$ of (3.6) will be independent with respect to $P_1, \dots, P_n$, as the formula clearly remains restricted and the dependency digraph remains unchanged. $\blacktriangleleft$

CLAIM 2 Given a formula $(A_i(P_1) \wedge B_i(P_1))$, satisfying the conditions of claim 1, phase 2 returns a formula $\exists P_1 \forall y[P_1(y) \vee \mathit{pref}((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k)) \wedge \mathit{pref}\, E]$ containing no Skolem functions, and with $\mathit{pref}((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))$ containing no predicate variables among $P_1, \ldots, P_n$.

PROOF OF CLAIM Recall that E is the formula $D' \wedge B_i(P_1)$. Since $A_i(P_1)$ contains no predicate variables among $P_2, \ldots, P_n$, it is clear that $\mathit{pref}((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))$ (and also D') will contain no predicate variables among $P_1, \ldots, P_n$. Recall that all top level $\exists$'s were pulled out in phase 1. Hence, in $A_i(P_1)$, P_1 does not occur in the scope of any $\exists$'s, since all positive occurrences are in $\exists\forall$-scope. It follows that, if quantifiers are pulled out as described in remark 3.2.4, then *pref* will contain only universal quantifiers. Nor are there any subformulae of the form $C \vee D$, with P_1 occurring both in C and D, in the conjunctive normal form obtained in step 1 (since this property is invariant under distribution of $\vee$ over $\wedge$). Hence no existential quantifiers will be introduced by step 2. It follows that no Skolem functions will be introduced in step 4. ◀

Combining claims 1 and 2, we see that $\mathit{pref}\, E$ remains independent with respect to $P_2, \ldots, P_n$ when $\mathit{pref}((x_1 \neq y \vee C_1) \wedge \cdots \wedge (x_k \neq y \vee C_k))$ is substituted for $\neg P$ in it, since no predicate variable is introduced into the scope of a universal quantifier. Lastly, since no Skolem functions occur, the deskolemization step is vacuous. QED

Theorem 3.3.9 *DLS succeeds in computing first-order equivalents for all formulae of the form $\exists P_1 \ldots \exists P_n A$ where A is independent with respect to $P_1, \ldots, P_n$.*

Proof. By induction along any linear order extending the partial order induced on the predicate variables $P_1, \ldots, P_n$ by the dependency digraph of A over $P_1, \ldots, P_n$, using lemma 3.3.8. QED

3.4 DLS on modal formulae

In this section we apply the results of the preceding sections to modal logic. Since DLS only eliminates existentially quantified predicate variables, the standard second-order translations of modal formulae have to be negated before they are given as input to the algorithm. Accordingly the negated results of successful executions will be (local) first-order equivalents for the original modal formulae. Hence, when we say that DLS succeeds on a modal formula φ, we mean that it succeeds on the input formula $\exists \overline{P} \neg \mathrm{ST}(\varphi, x)$. [Sza93] contains many examples of such executions of DLS on modal formulae. The following lemma is easy to prove.

Lemma 3.4.1 *Let $\varphi \in \mathcal{L}_\tau$ be in negation normal form. Then*

1. $\mathrm{ST}(\varphi, x)$ *is standardized (modulo associativity) with respect to all predicate symbols in* $\mathrm{ST}(\mathsf{PROP}(\varphi))$*,*
2. $\mathrm{ST}(\varphi, x)$ *is independent with respect to* $\mathrm{ST}(\mathsf{PROP}(\varphi))$ *whenever φ is the negation of a Sahlqvist or inductive formula written in negation normal form.*

The next theorem is now an immediate consequence of lemma 3.4.1 and theorem 3.3.9.

Theorem 3.4.2 *DLS succeeds in computing the first-order frame correspondent of all Sahlqvist and Inductive formulae.*

Next we turn to the question of the canonicity of the modal formulae reducible by DLS. We fix for the rest of this section a modal language $\mathcal{L}_\pi$, with π a unary modal similarity type. The notions of *benign*, *malignant* and *non-malignant* occurrences (with respect to propositional variables) of conjunctions and disjunctions in $\mathcal{L}_\pi$-formulae, are the obvious modal analogues of these definitions for predicate logic formulae.

Definition 3.4.3 An occurrence of a propositional variable p, is in $\Box\Diamond$-*scope* in an $\mathcal{L}_\pi$-formula φ if, in φ, it is not in the scope any

(i) box which is in the scope of a diamond, or

(ii) non-benign (w.r.t. p) conjunction within the scope of a diamond, or

(iii) malignant disjunction (w.r.t. p) within the scope of a diamond.

Definition 3.4.4 A propositional variable p is in *good scope* in an $\mathcal{L}_\pi$-formula φ, if

(i) no disjunction or conjunction malignant with respect p occurs within the scope of a diamond in φ, and

(ii) φ contains no subformula of the form $\psi_1 \vee \psi_2$ where ψ_1 and ψ_2 contain occurrences of p out of $\Box\Diamond$-scope of opposite polarities.

Lemma 3.4.5 *Let $\varphi \in \mathcal{L}_\pi$, and let φ' be the result of rewriting $\neg\varphi$ in negation normal form. Then P, corresponding to p, is in good scope in* $\mathrm{ST}(\varphi', x)$ *if and only if p is in good scope in φ.*

The next theorem is a direct consequence of lemma 3.4.5 and theorem 3.3.3.

Theorem 3.4.6 *Suppose $\varphi \in \mathcal{L}_\pi$ contains exactly one propositional variable, say p. Then DLS will succeed in computing a first-order correspondent for φ iff p is in good scope in φ.*

Using a strategy similar to that used to prove theorem 2.6.3, it is not difficult to show that SQEMA succeeds on all $\mathcal{L}_\pi$-formulae in which the only occurring propositional variable is in good scope. Combining this fact with theorem 3.4.6 and the canonicity theorem for SQEMA (theorem 2.5.23) we obtain:

Theorem 3.4.7 *Suppose $\varphi \in \mathcal{L}_\pi$ contains exactly one propositional variable. Then DLS succeeds in computing a first-order correspondent for φ only if φ is canonical.*

We conclude this section with an example illustrating some aspects of the execution of DLS on modal formulae.

Example 3.4.8 Consider the Sahlqvist formula $\Diamond((p \wedge \neg q) \vee \Box p) \wedge q \to p$. Negated this becomes $\Diamond((p \wedge \neg q) \vee \Box p) \wedge q \wedge \neg p$. Translated the latter becomes

$$\exists P \exists Q(\exists z_1(Rx_0z_1 \wedge ((P(z_1) \wedge \neg Q(z_1)) \vee \forall z_2(\neg Rz_1z_2 \vee P(z_2)))) \wedge Q(x_0) \wedge \neg P(x_0)). \quad (3.11)$$

Suppose we eliminate P first. All that happens in phase 1 is that the top level existential quantifier is pulled out, yielding

$$\exists z_1 \exists P \exists Q((Rx_0z_1 \wedge ((P(z_1) \wedge \neg Q(z_1)) \vee \forall z_2(\neg Rz_1z_2 \vee P(z_2)))) \wedge Q(x_0) \wedge \neg P(x_0)). \quad (3.12)$$

Fase 1 succeeds, as the matrix of (3.12) has the form $A(P) \wedge B(P)$ with $A(P)$ positive and $B(P)$ negative in P. Specifically $A(P)$ is $(Rx_0z_1 \wedge ((P(z_1) \wedge \neg Q(z_1)) \vee \forall z_2(\neg Rz_1z_2 \vee P(z_2))))$ while $B(P)$ is $Q(x_0) \wedge \neg P(x_0)$. Step 1 of phase 2 transforms $A(P)$ into

$$\forall z_2(Rx_0z_1 \wedge (P(z_1) \vee \neg Rz_1z_2 \vee P(z_2)) \wedge (\neg Q(z_1) \vee \neg Rz_1z_2 \vee P(z_2))). \quad (3.13)$$

Steps 2 and 3 of phase 2 transform the second two conjuncts of the matrix of (3.13) respectively into:

$$\exists x_1(\forall y(P(y) \vee x_1 \neq y \vee C_1) \wedge (x_1 = z_1 \vee x_1 = z_2))$$

with C_1 the formula $\neg Rz_1z_2$, and

$$\forall y(P(y) \vee y \neq z_2 \vee C_2)$$

with C_2 the formula $\neg Q(z_1) \vee \neg Rz_1z_2$. Step 4 of phase 2 now yields the formula

$$\exists z_1 \exists f \exists P \forall z_2(\forall y(P(y) \vee f(z_2) \neq y \vee C_1) \wedge \forall y(P(y) \vee y \neq z_2 \vee C_2) \wedge E) \quad (3.14)$$

where E is the formula $Rx_0z_1 \wedge (f(z_2) = z_1 \vee f(z_2) = z_2) \wedge (Q(x_0) \wedge \neg P(x_0))$. Step 5 of phase 2 transforms (3.14) into

$$\exists z_1 \exists f \exists P \forall y((P(y) \vee \forall z_2((f(z_2) \neq y \vee C_1) \wedge (y \neq z_2 \vee C_2)) \wedge \forall z_2 E) \quad (3.15)$$

In step 1 of phase 3 we apply Ackermann's lemma to obtain from (3.15)

$$\exists z_1 \exists f \forall z_2(Rx_0z_1 \wedge (f(z_2) = z_1 \vee f(z_2) = z_2) \wedge (Q(x_0) \wedge \forall z_2((f(z_2) \neq x_0 \vee C_1) \wedge (x_0 \neq z_2 \vee C_2)))) \quad (3.16)$$

Deskolemization yields

$$\exists z_1 \forall z_2 \exists u(Rx_0z_1 \wedge (u = z_1 \vee u = z_2) \wedge (Q(x_0) \wedge \forall z_2((u \neq x_0 \vee C_1) \wedge (x_0 \neq z_2 \vee C_2)))) \quad (3.17)$$

i.e.

$$(3.18) \quad \begin{array}{ll} & \exists z_1 \forall z_2 \exists u(Rx_0z_1 \wedge (u = z_1 \vee u = z_2) \\ \wedge & (Q(x_0) \wedge \forall z_2((u \neq x_0 \vee \neg Rz_1z_2) \wedge (x_0 \neq z_2 \vee \neg Q(z_1) \vee \neg Rz_1z_2)))) \end{array}$$

No simplification rules are applicable. Now to eliminate Q we return to phase 1. It is should be clear that this will fail to achieve the desired form separated for Q — this is made impossible by the presence of the $\forall z_2 \exists u$ in the quantifier prefix.

Notice how the modal structure of the formula is lost in the of the forgoing computation — we start of with the translation of a modal formula in (3.11) and end up with a formula (3.18) which is very difficult to see as (being equivalent to) the translation of a modal or hybrid formula.

However, had we started by 'bubbling up disjunctions', as is done in the strategy employed in the proof of lemma 3.3.8, the picture would have looked very different, as we will briefly outline: $(\Diamond(p \wedge \neg q) \wedge q \wedge \neg p) \vee (\Diamond\Box p \wedge q \wedge \neg p)$ translates into two disjunctions

$$\exists P \exists Q[\exists z_1(Rx_0z_1 \wedge P(z_1) \wedge \neg Q(z_1)) \wedge Q(x_0) \wedge \neg P(x_0)] \tag{3.19}$$

and

$$\exists P \exists Q[\exists z_3(Rx_0z_3 \wedge \forall z_4(\neg Rz_3z_4 \vee P(z_4))) \wedge Q(x_0) \wedge \neg P(x_0)] \tag{3.20}$$

on which DLS proceeds separately. We will only consider the execution on the first disjunct:

Phase 1:

$$\exists z_1 \exists P \exists Q[(Rx_0z_1 \wedge P(z_1) \wedge \neg Q(z_1)) \wedge Q(x_0) \wedge \neg P(x_0)]$$

Phase 2:

$$\exists z_1 \exists P \exists Q \forall y((P(y) \vee (y \neq z_1)) \wedge (\neg Q(z_1) \wedge Rx_0z_1 \wedge Q(x_0) \wedge \neg P(x_0)))$$

Phase 3:

$$\exists z_1 \exists Q((\neg Q(z_1) \wedge Rx_0z_1 \wedge Q(x_0) \wedge x_0 \neq z_1))$$

Note that we can still encode the resulting second-order statement, à la SQEMA, as the global satisfiability of the hybrid formula

$$(\mathbf{i} \to \Diamond\mathbf{j} \wedge q) \wedge (\mathbf{j} \to (\neg\mathbf{i} \wedge \neg q))$$

subject to the constraint that $\mathbf{i}$ be interpreted as x_0. Now eliminating Q, phase 1 is vacuous, while phase 2 yields

$$\exists z_1 \exists Q \forall y((Q(y) \vee y \neq x_0) \wedge (\neg Q(z_1) \wedge Rx_0z_1 \wedge x_0 \neq z_1)).$$

Phase 3 yields

$$\exists z_1(z_1 \neq x_0 \wedge Rx_0z_1 \wedge x_0 \neq z_1),$$

which is again expressible as the global satisfiability of the pure hybrid formula

$$\mathbf{i} \to \Diamond(\mathbf{j} \wedge \neg\mathbf{i})$$

with $\mathbf{i}$ interpreted as x_0. ◁

3.5 Conclusion and open questions

We have delineated, within certain limits and given certain assumptions, the second-order and modal formulae on which DLS succeeds. We were able to obtain a syntactic characterization theorem only in the case of formulae in a single predicate or propositional variable. This is not surprising, as DLS is almost certainly too powerful for the full class of formulae which it can reduce to admit of a convenient syntactic characterization.

Applying these results to the translations of modal formulae, we saw that DLS is powerful enough to reduce all Sahlqvist and Inductive formulae. Moreover, all modal formulae in a single propositional variable on which DLS succeeds are canonical. In closing we mention some further directions and conjectures.

A more thoroughgoing comparison between DLS and SQEMA must be made. It seems possible, under certain circumstances, to rewrite the formulae obtained during the execution of DLS as formulae in the modal language extended with inverse modalities and nominals. This is the language in which SQEMA works. In so doing one might be able to simulate each algorithm with the other. An example of a formula on which SQEMA succeeds, but on which DLS fails, was given in examples 2.3.3 and 3.2.10. However, SQEMA's success on this formula depends crucially on its ability to perform some basic propositional reasoning. Suppose the algorithm SQEMA^- were obtained by removing SQEMA's auxiliary rules, then we make the following conjecture:

Conjecture 3.5.1 *DLS and SQEMA^- are equivalent in terms of the modal formulae reducible by them.*

This strategy of translation and simulation also seems the most plausible route to a general canonicity theorem for DLS, the chief difficulty with which seems to be caused by the progressive loss of the (modal) structure of the original input formula which sometimes occurs (as illustrated in example 3.4.8) as DLS is iterated on it. In example 3.4.8 it was seen that this loss of structure might be forestalled by some suitable preprocessing, but it is not clear whether this is always achievable.

Conjecture 3.5.2 *All modal formulae on which DLS succeeds are canonical.*

Chapter 4

Polyadic Languages

In this chapter we extend the algorithm SQEMA, introduced in the previous chapter, to polyadic and reversive languages. The adaptation of the algorithm itself could not be more straight-forward. The proof that all formulae on which the adapted algorithm now succeeds are canonical, however, involves some non-trivial complications. We are also now able to introduce the full class of inductive formulae, and to show how SQEMA succeeds in finding first-order equivalents for these, as well as in proving their canonicity. This chapter is adapted from sections of [CGV06b].

4.1 Reversive polyadic languages and logics

Until now we have only dealt with monadic modal languages. In this section we properly introduce polyadic languages and some of the technicalities that accompany them. This section is based on the exposition of these matters given in [GV06]. The motivation for the way we treat polyadic languages here, is a need for a more complete, interrelated set of modalities. For example, instead of having to write $\langle\alpha\rangle(p, \langle\beta\rangle(q, \neg p \wedge q))$ we would like to be able to write $\langle\alpha(\iota_1, \beta(\iota_1, \iota_2))\rangle(p, q, \neg p, q))$. In other words, not only would we like to be able to compose modalities, but we also want to be able to do away with conjunction and disjunction in favour of modalities. We further want to be able to take the inverses of the resulting modalities, which again have to be composable in their turn. Having such a rich store of modal operators at our disposal will allow us, amongst other things, to define syntactic classes very simply and succinctly.

4.1.1 Polyadic similarity types

As before modal *similarity type* $\tau = (O, \rho_0)$ consists of a nonempty set O of *basic modal terms*, together with an *arity function* $\rho_0 : O \to \omega$ assigning to each modal term $\alpha \in O$ a natural number $\rho_0(\alpha)$. We will assume that τ contains a 0-ary modal term $\perp$, a unary one ι_1, and a binary one ι_2. As will become clear from the semantics below, the special modal term $\perp$ will be interpreted as falsum, ι_1 as the self-dual identity, ι_2 as $\wedge$, and its dual as $\vee$. Treating these connectives as modalities will enable us to define a more general class of polyadic inductive formulae (see e.g. [GV06]).

Definition 4.1.1 Given a modal similarity type τ and a (fixed) set of proposition letters Θ, we define by simultaneous mutual induction the set of *polyadic modal terms* MT_τ and their *arity function* ρ extending ρ_0, and the set of *polyadic modal formulae* of the language $\mathcal{L}_\tau(\Theta)$ as follows:

(MT1) Every basic modal term from O is a modal term of the predefined arity.

(MT2) Every $\mathcal{L}_\tau$-formula containing no variables (variable-free or constant formula) is a 0-ary modal term.

(MT3) If $n > 0$, α, $\beta_1, \ldots, \beta_n \in MT_\tau$ and $\rho(\alpha) = n$, then $\alpha(\beta_1, \ldots, \beta_n) \in MT_\tau$ and $\rho(\alpha(\beta_1, \ldots, \beta_n)) = \rho(\beta_1) + \cdots + \rho(\beta_n)$.

Modal terms of arity 0 will be called *modal constants*.

(MF1) Every propositional variable is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF2) Every 0-ary modal term in MT_τ is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF3) If φ is a formula of $\mathcal{L}_\tau(\Theta)$ then $\neg\varphi$ is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF4) If φ and ψ are formulae of $\mathcal{L}_\tau(\Theta)$ then $\varphi \vee \psi$ is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF5) If φ and ψ are formulae of $\mathcal{L}_\tau(\Theta)$ then $\varphi \wedge \psi$ is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF6) If $A_1, \ldots, A_n$ are formulae of $\mathcal{L}_\tau(\Theta)$, α a modal term and $\rho(\alpha) = n > 0$, then $\langle\alpha\rangle(A_1, \ldots, A_n)$ is a formula of $\mathcal{L}_\tau(\Theta)$.

(MF7) If $A_1, \ldots, A_n$ are formulae of $\mathcal{L}_\tau(\Theta)$, α a modal term and $\rho(\alpha) = n > 0$, then $[\alpha](A_1, \ldots, A_n)$ is a formula of $\mathcal{L}_\tau(\Theta)$.

As will be seen further in subsection 4.1.2, $\wedge$ and $\vee$ may be eliminated in favour of $\langle\iota_2\rangle$ and $[\iota_2]$, respectively, making clauses (MF4) and (MF5) redundant. However, for technical convenience we choose to retain $\wedge$ and $\vee$ as primitive connectives of the language. The connectives $\rightarrow$ and $\leftrightarrow$ are defined in the usual. Adding the clause

(MT6) If α is a modal term of arity $n > 0$ then $\alpha^{-1}, \ldots, \alpha^{-n}$ are modal terms of arity n.

to definition 4.1.1, we obtain the set of terms $\mathrm{MT}_{r(\tau)}$, called the *complete reversive extension of* MT_τ. Similarly, the corresponding language $\mathcal{L}_{r(\tau)}$ is called the *complete reversive extension of* $\mathcal{L}_\tau$. A set of modal terms MT_τ will be called *completely reversive*, or simply *reversive*, if $\mathrm{MT}_{r(\tau)} = \mathrm{MT}_\tau$. Accordingly, languages over a completely reversive sets of terms will be called *completely reversive*, or simply *reversive*. If one were to opt for the following, weaker version of (MT6)

(MT6′) If α is a modal term from MT_τ of arity $n > 0$ then $\alpha^{-1}, \ldots, \alpha^{-n}$ are modal terms of arity n.

one would obtain the set of modal terms $\mathrm{MT}_{\tau r}$, which we will call the *partial reversive extension* of MT_τ. Accordingly the language $\mathcal{L}_{\tau r}$ is the *partial reversive* extension of $\mathcal{L}_\tau$.[1] Hence $\mathrm{MT}_{\tau r}$ is not necessarily closed under inverses, but only under inverses of terms in MT_τ. So $\mathcal{L}_{\tau r}$ is a sublanguage of $\mathcal{L}_{\tau r}$, yet, modulo the permutation of arguments of modal operators, they are equally expressive, e.g.

$$\langle(\alpha^{-i})^{-j}\rangle(A_1,\ldots,A_n) \equiv_{\mathrm{sem}} \langle\alpha^{-j}\rangle(A_1,\ldots,A_{i-1},A_j,A_{i+1},\ldots,A_{j-1},A_i,A_{j+1},\ldots,A_n)$$

when $i<j$. Even though we will not be concerned with partial reversive extensions further in this chapter, it is important to take note of them for the sake of comparison with other results in the literature.

For technical purposes we extend the series of ι's with n-ary modalities ι_n inductively as follows: $\iota_{n+1} = \iota_2(\iota_1,\iota_n)$ for $n>1$.

4.1.2 Semantics

As far the semantics of $\mathcal{L}_\tau$ and $\mathcal{L}_\tau$ is concerned, there are no surprises. We therefore only specify the Kripke structures upon which the models used for interpreting the langauge are based. Given a modal similarity type τ, a *(Kripke) τ-frame* is, as usual, a structure $\mathfrak{F} = (W,\{R_\alpha\}_{\alpha\in\tau})$, consisting of a non-empty set W of *possible worlds* and, for each modal term $\alpha\in\tau$, a $(\rho(\alpha)+1)$-ary *accessibility relation* between possible worlds $R_\alpha \subseteq W^{\rho(\alpha)+1}$. The relations associated with special modal terms are fixed: $R_{\iota_1} = \{(w,w) \mid w\in W\}$, $R_{\iota_2} = \{(w,w,w) \mid w \in W\}$. The (unary) relation associated with a variable-free formula is simply the set of states where that formula is true, given by the standard semantics.

The relation associated with a composite modal term is defined as follows: Suppose $\rho(\alpha) = n$ and $\rho(\beta_i) = b_i$, for $1\le i\le n$. Then $R_{\alpha(\beta_1,\ldots,\beta_n)}$ consists of all tuples of elements of W of the from $(w,w_{11},\ldots,w_{1b_1},\ldots,w_{n1},\ldots,w_{nb_n})$, for which there exist $u_1\ldots u_n\in W$ such that $R_\alpha w u_1\ldots u_n$ and $R_{\beta_i} u_i w_{i1}\ldots w_{ib_i}$ for each $1\le i\le n$.

The relations for inverses are given as expected: For any $\alpha\in\mathrm{MT}_{r(\tau)}$, $1\le j\le\rho(\alpha)$, and $w,v_1,\ldots,\ldots,v_{\rho(\alpha)}\in W$, we declare that

$$R_\alpha^{-j}(w,v_1,\ldots,\ldots,v_{\rho(\alpha)}) \text{ iff } R_\alpha(v_j,v_1,\ldots,v_{j-1},w,v_{j+1},\ldots,v_{\rho(\alpha)}).$$

Note that, by our definition, a Kripke τ-frame only contains relations associated with the *basic* modal terms $\alpha\in\tau$. However, these relations completely determine the relations that are to be associated with the terms in the full set $\mathrm{MT}_{r(\tau)}$. Notions of *model*, *truth*, *validity* and the like are defined as usual.

Given the fixed relations associated with the special modal terms $\iota_1,\iota_2,\ldots$, we have that

$$\langle\iota_1\rangle\varphi \equiv_{\mathrm{sem}} [\iota_1]\varphi \equiv \varphi,$$

$$\langle\iota_n\rangle(\varphi_1,\ldots,\varphi_n) \equiv_{\mathrm{sem}} \varphi_1\wedge\cdots\wedge\varphi_n \quad\text{and}\quad [\iota_n](\varphi_1,\ldots,\varphi_n) \equiv_{\mathrm{sem}} \varphi_1\vee\cdots\vee\varphi_n.$$

This makes it possible, when writing $\mathcal{L}_\tau$ and $\mathcal{L}_{r(\tau)}$-formulae, to do away with all boolean connectives other than negation — a fact that will be exploited subsequently.

[1] In [GV06] and [CGV06b] $\mathrm{MT}_{\tau r}$ and $\mathcal{L}_{\tau r}$ are called the *reversive extensions* of MT_τ and $\mathcal{L}_\tau$.

With each similarity type τ we associate a first-order language L_0^τ containing $=$ and a $(\rho(\alpha)+1)$-ary relation symbol R_α for each basic modal term $\alpha \in \tau$, other that $\bot, \iota_1$ and ι_2. The language L_1^τ extends L_0^τ with unary predicate symbols $P_1, P_2, \ldots$ corresponding to the propositional variables $p_1, p_2, \ldots$. The standard translation function $\mathrm{ST}(\cdot,\cdot)$ is extended in the obvious way to translate every $\mathcal{L}_{r(\tau)}$-formula into an L_1^τ-formula such $(\mathcal{M}, m) \Vdash \varphi$ iff $\mathcal{M} \models \mathrm{ST}(\varphi, x)[x := m]$ for every pointed τ-model $(\mathcal{M}, m)$ and $\mathcal{L}_{r(\tau)}$-formula φ.

4.1.3 Permutations versus inverses

To facilitate some proofs below, we introduce the following syntactic shorthand: Let $\alpha \in \mathrm{MT}_\tau$ with $\rho(\alpha) = n$, and let σ be a permutation of $\{0, 1, \ldots, n\}$, i.e. a bijection from $\{0, 1, \ldots, n\}$ onto itself. Then we admit α^σ as a modal term. Let

$$R_\alpha^\sigma = \{(x_0, x_1, \ldots, x_n) \mid R_\alpha x_{\sigma(0)}, x_{\sigma(1)}, \ldots, x_{\sigma(n)}\}.$$

Given a permutation σ, its inverse will be denoted by $\overline{\sigma}$. We will identify modal terms $\alpha^{\overline{\overline{\sigma}}}$ and α without further ado. Note that $R_\alpha y_0, y_1, \ldots, y_n$ iff $R_\alpha^\sigma y_{\overline{\sigma}(0)}, y_{\overline{\sigma}(1)}, \ldots, y_{\overline{\sigma}(n)}$. The semantics of the corresponding language is what is to be expected, i.e. $(\mathcal{M}, m) \Vdash \langle \alpha^\sigma \rangle(\varphi_1, \ldots, \varphi_n)$ iff there are $m_1, \ldots, m_n \in \mathcal{M}$ such that $R_\alpha^\sigma m, m_1, \ldots, m_n$ and $(\mathcal{M}, m_i) \Vdash \varphi_i$ for all $1 \leq i \leq n$. Since every permutation is obtainable by the repeated swapping of positions, the language so obtained and $\mathcal{L}_{r(\tau)}$ will be equally expressive. To be precise, let a *transposition* be a permutation σ such that, for some $0 \leq i < j \leq n$ we have $\sigma(i) = j$ and $\sigma(j) = i$ and $\sigma(k) = k$ for all $i \neq k \neq j$. An easy induction on n shows that every permutation of $\{0, 1, \ldots, n\}$ can be obtained as a composition of such transpositions. Now suppose σ is a transposition which swaps i and j, and that $R_\alpha^\sigma x_0 x_1 \ldots x_n$. This is the case iff $R_\alpha x_0 x_1 \ldots x_{i-1} x_j x_{i+1} \ldots x_{j-1} x_i x_{j+1} \ldots x_n$ iff $((R_\alpha^{-i})^{-j})^{-i} x_0 x_1 \ldots x_{i-1} x_i x_{i+1} \ldots x_{j-1} x_j x_{j+1} \ldots x_n$. It follows that for any n-ary modal term α and permutation σ of $\{0, 1 \ldots, n\}$, the term α^σ is equivalent to a term which can be obtained from α by applying inverses and compositions.

The next lemma generalizes the fact enshrined as axiom R1 in [GV06].

Lemma 4.1.2 *Let α be a modal term with $\rho(\alpha) = n$, $A, B_1, \ldots, A_n$ any formulae, and σ a permutation of $\{0, 1, \ldots, n\}$ with $\sigma(0) = k \neq 0$ and $\overline{\sigma}(0) = j \neq 0$. Then*

$$\Vdash A \to [\alpha^\sigma](\neg B_{\overline{\sigma}(1)}, \ldots, \neg B_{\overline{\sigma}(k-1)}, C, \neg B_{\overline{\sigma}(k+1)}, \ldots, \neg B_{\overline{\sigma}(n)})$$

where C is the formula

$$\langle \alpha \rangle (B_1, \ldots, B_{j-1}, A, B_{j+1}, \ldots, B_n).$$

Proof. Let $(\mathcal{M}, m_0)$ be any pointed model such that $(\mathcal{M}, m_0) \Vdash A$ and suppose, for the sake of contradiction, that

$$(\mathcal{M}, m) \nVdash [\alpha^\sigma](\neg B_{\overline{\sigma}(1)}, \ldots, \neg B_{\overline{\sigma}(k-1)}, C, \neg B_{\overline{\sigma}(k+1)}, \ldots, \neg B_{\overline{\sigma}(n)}),$$

i.e. there are $m_1, \ldots, m_n \in \mathcal{M}$ be such that $R_\alpha^\sigma m_0, m_1, \ldots m_n$ and, for all $i \neq 0$ and $i \neq k$ we have $(\mathcal{M}, m_i) \Vdash B_{\overline{\sigma}(i)}$ (or, equivalently, for all $i \neq 0$ and $i \neq j$ we have $(\mathcal{M}, m_{\sigma(i)}) \Vdash B_i$), while and $(\mathcal{M}, m_k) \Vdash \neg C$. That is, $(\mathcal{M}, m_k) \Vdash [\alpha](\neg B_1, \ldots, \neg B_{j-1}, \neg A, \neg B_{j+1}, \ldots, \neg B_n)$.

But then $R_\alpha m_{\sigma(0)}, m_{\sigma(1)}, \ldots, m_{\sigma(j-1)}, m_{\sigma(j)}, m_{\sigma(j+1)}, \ldots m_{\sigma(n)}$, i.e. $R_\alpha m_k, m_{\sigma(1)}, \ldots, m_{\sigma(j-1)}, m_0, m_{\sigma(j+1)}, \ldots m_{\sigma(n)}$. As we have already remarked, for all $i \neq 0$ and $i \neq j$ we have $(\mathcal{M}, m_{\sigma(i)}) \Vdash B_i$, hence we are forced to conclude that $(\mathcal{M}, m_0) \Vdash \neg A$, contradicting our original assumption that $(\mathcal{M}, m_0) \Vdash A$. QED

Having introduced polyadic languages properly, we are now able to define the polyadic inductive formulae of [GV02] and [GV06]. For the rest of this section we fix an arbitrary (polyadic) modal similarity type τ, and work in the language $\mathcal{L}_{r(\tau)}$ — when speaking of formulae, we will always mean $\mathcal{L}_{r(\tau)}$-formulae, unless otherwise indicated.

4.2 Polyadic inductive formulae

A formula $[\beta](N_1, \ldots, N_m)$, where β is any m-ary modal term from $\mathrm{MT}_{r(\tau)}$ and $N_1, \ldots, N_m$ are negative formulae, will be called a *headless box formula* (or simply a *headless box*). A formula of the form $[\beta](p, N_1, \ldots, N_m)$, where $\beta \in \mathrm{MT}_{r(\tau)}$ is an $(m+1)$-ary modal term, p is a propositional variable, and $N_1, \ldots, N_m$ are negative formulae, will be called a *headed box formula* (or *headed box*) with *head* p. (The head of a headed box need in fact not occur as the first argument of the box-operator — we merely write it as such for the sake of simplicity and uniformity. As the reader can readily verify, nothing that follows changes in any essential way if we drop this convention.) The occurrence of a variable as the head of a box formula is called an *essential occurrence*, while all other variable occurrences in (headed or headless) box formulae are called *inessential*. A *box formula* is either a headed or headless box formula.

Definition 4.2.1 A *regular formula* is a formula of the form $[\alpha](\neg B_1, \ldots, \neg B_n)$, where α is an n-ary modal term and $B_1, \ldots, B_n$ are box formulae.

Definition 4.2.2 The *dependency digraph* of a regular formula $A = [\alpha](\neg B_1, \ldots, \neg B_n)$ is the digraph $G_A = \langle V_A, E_A \rangle$. The vertex set V_A is the set $\{p_1, \ldots, p_m\}$ of all heads of headed boxes among $B_1, \ldots, B_n$. The edge set $E_A \subseteq V_A \times V_A$ is such that $(p_i, p_j) \in E_A$ iff p_i occurs inessentially in some $B_1, \ldots, B_n$ with head p_j. A digraph is *acyclic* when it contains no directed cycles or loops.

Definition 4.2.3 An *inductive formula* is any regular formula with an acyclic dependency digraph.

Inductive formulae were originally referred to as 'polyadic Sahlqvist formulae' in [GV02], where their elementarity and local d-persistence were also proved.

For a formula to be inductive then, it has to be of a very specific syntactic shape. However, the definition is not as restrictive as it may seem at first. By composing modalities and using the special ι-modalities, many formulae may be equivalently rewritten as inductive formulae. For example, we may rewrite any monadic box formula

$$A_1 \to [\alpha_1](\cdots [\alpha_k](A_{k+1} \to p) \cdots),$$

with $\rho(\alpha_1) = \cdots = \rho(\alpha_k) = 1$, as

$$[\iota_2(\iota_1, \alpha_1(\cdots \alpha_k(\iota_2)\cdots))](\neg A_1, \ldots, \neg A_{k+1}, p).$$

Using this fact, the following proposition is easily proven:

Proposition 4.2.4 *Every monadic inductive formula is semantically equivalent to a conjunction of inductive formula which may be effectively obtained from it.*

So, modulo semantic equivalence, or even modulo preprocessing, the polyadic inductive formulae subsume the monadic inductive, and hence the Sahlqvist, formulae. Hence, in the light of corollary 1.3.19, the inductive formulae also subsume the Sahlqvist–van Benthem formulae, modulo local equivalence.

Example 4.2.5 Let α be a monadic modal term, then

$$p \wedge [\alpha](\langle\alpha\rangle p \to [\alpha]q) \to \langle\alpha\rangle[\alpha][\alpha]q$$

is the monadic inductive formula D from example 1.3.17. First rewriting as

$$\neg p \vee \neg[\alpha]([\alpha]\neg p \vee [\alpha]q) \vee \neg[\alpha]\langle\alpha\rangle\langle\alpha\rangle q,$$

and then composing modalities and replacing disjunctions we can rewrite this formula equivalently as the inductive formula

$$[\iota_3](\neg[\iota_1]p, \neg[\alpha(\iota_2(\alpha,\alpha))](q,\neg p), \neg[\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg q)).$$

$\lhd$

4.3 Extending SQEMA

To enable SQEMA to deal with polyadic languages, we have to add some new transformation rules to its repertoire. The only new rules that are needed fall in the categories *rules for logical connectives* and *normalization rules*. Apart from these additions the algorithm itself as well as the Ackermann, polarity switching and auxiliary rules remain unchanged. Specifically, the set of rules for the logical connectives is adapted as follows:

1. The $\wedge$-rule and the left and right-shift $\vee$-rules remain unchanged.

2. The $\Box$-rule is subsumed by the following *polyadic box rule*:

$$\frac{A \Rightarrow [\alpha](B_1, \ldots, B_n)}{\langle\alpha^{-k}\rangle(\neg B_1, \ldots, \neg B_{k-1}, A, \neg B_{k+1}, \ldots \neg B_n) \Rightarrow B_k} \qquad \text{(Polyadic } \Box\text{-rule)}$$

   for $\alpha \in \mathrm{MT}_{r(\tau)}$ any modal term of arity n, and any $1 \leq k \leq n$.

3. The inverse $\Diamond$-rule is subsumed by the following *polyadic inverse diamond rule*:

$$\frac{\langle \alpha^{-k} \rangle (B_1, \ldots, B_n) \Rightarrow A}{B_k \Rightarrow [\alpha](\neg B_1, \ldots, \neg B_{k-1}, A, \neg B_{k+1}, \ldots, \neg B_n)} \qquad \text{(Polyadic inverse } \Diamond\text{-rule)}$$

   for $\alpha \in \mathrm{MT}_{r(\tau)}$ any modal term of arity n, and any $1 \leq k \leq n$.

4. The $\Diamond$-rule is subsumed by the following *polyadic diamond rule*:

$$\frac{\mathbf{j} \Rightarrow \langle \alpha \rangle (B_1, \ldots, B_n)}{\mathbf{j} \Rightarrow \langle \alpha \rangle (\mathbf{k}_1, \ldots, \mathbf{k}_n), \mathbf{k}_1 \Rightarrow B_1, \ldots, \mathbf{k}_n \Rightarrow B_n} \qquad \text{(Polyadic } \Diamond\text{-rule)}$$

   for $\alpha \in \mathrm{MT}_{r(\tau)}$ any modal term of arity n, and any new nominals, $\mathbf{k}_1, \ldots, \mathbf{k}_n$ not appearing in the system.

As far as the normalization rules are concerned, the rules for the duality of boxes and diamonds are subsumed by the obvious polyadic analogues:

1. Replace $\neg\langle \alpha \rangle (\varphi_1, \ldots, \varphi_n)$ with $[\alpha](\neg\varphi_1, \ldots, \neg\varphi_n)$, for any term $\alpha \in \mathrm{MT}_{r(\tau)}$;

2. Replace $\neg[\alpha](\varphi_1, \ldots, \varphi_n)$ with $\langle \alpha \rangle (\neg\varphi_1, \ldots, \neg\varphi_n)$, for any term $\alpha \in \mathrm{MT}_{r(\tau)}$.

We note that, whereas in the monadic case the only rule for logical connectives that could lead from a normalized to a non-normalized sequent was the $\vee$-rule, in the polyadic case normality can also be lost by the application of the $\Box$ and inverse $\Diamond$-rules.

4.4 Examples

In this section we illustrate the newly extended algorithm by means of a few examples.

Example 4.4.1 Consider the formula

$$D = [\iota_3](\neg[\iota_1]p, \neg[\alpha(\iota_2(\alpha, \alpha))](q, \neg p), \neg[\alpha](\langle \alpha \rangle \langle \alpha \rangle \neg q)),$$

of example 4.2.5. Here is how SQEMA computes it local first-order frame equivalent:

Phase 1 Preprocessing yields $\langle \iota_3 \rangle ([\iota_1]p, [\alpha(\iota_2(\alpha, \alpha))](q, \neg p), [\alpha](\langle \alpha \rangle \langle \alpha \rangle \neg q))$.

Phase 2 There is only one initial system, namely

$$\| \mathbf{i} \Rightarrow \langle \iota_3 \rangle ([\iota_1]p, [\alpha(\iota_2(\alpha, \alpha))](q, \neg p), [\alpha](\langle \alpha \rangle \langle \alpha \rangle \neg q)) \ .$$

Applying the $\Diamond$-rule yields:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \iota_3 \rangle (\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ \mathbf{j}_1 \Rightarrow [\iota_1]p \\ \mathbf{j}_2 \Rightarrow [\alpha(\iota_2(\alpha, \alpha))](q, \neg p) \\ \mathbf{j}_3 \Rightarrow [\alpha](\langle \alpha \rangle \langle \alpha \rangle \neg q) \end{array} \right. .$$

We choose to eliminate p first, and apply the $\Box$-rule to the second sequent:

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \langle\iota_1^{-1}\rangle\mathbf{j}_1 \Rightarrow p\\ \mathbf{j}_2 \Rightarrow [\alpha(\iota_2(\alpha,\alpha))](q, \neg p)\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg q)\end{array}\right. .$$

The system is now ready for the application of the Ackermann-rule, to eliminate p:

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \mathbf{j}_2 \Rightarrow [\alpha(\iota_2(\alpha,\alpha))](q, \neg\langle\iota_1^{-1}\rangle\mathbf{j}_1)\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg q)\end{array}\right. ,$$

which, after re-normalization, becomes

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \mathbf{j}_2 \Rightarrow [\alpha(\iota_2(\alpha,\alpha))](q, [\iota_1^{-1}]\neg\mathbf{j}_1)\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg q)\end{array}\right. .$$

Only q remains to be eliminated. To that aim, we apply the $\Box$-rule to the second sequent, obtaining

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \langle(\alpha(\iota_2(\alpha,\alpha)))^{-1}\rangle(\mathbf{j}_2, \neg[\iota_1^{-1}]\neg\mathbf{j}_1) \Rightarrow q\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg q)\end{array}\right. .$$

Applying Ackermann-rule again eliminates q and yields

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle\neg\langle(\alpha(\iota_2(\alpha,\alpha)))^{-1}\rangle(\mathbf{j}_2, \neg[\iota_1^{-1}]\neg\mathbf{j}_1))\end{array}\right. ,$$

which re-normalization turns into

$$\left\|\begin{array}{l}\mathbf{i} \Rightarrow \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)\\ \mathbf{j}_3 \Rightarrow [\alpha](\langle\alpha\rangle\langle\alpha\rangle[(\alpha(\iota_2(\alpha,\alpha)))^{-1}](\neg\mathbf{j}_2, [\iota_1^{-1}]\neg\mathbf{j}_1))\end{array}\right. .$$

Phase 3 We thus obtain

$$\mathsf{pure}(D) = (\neg\mathbf{i} \vee \langle\iota_3\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)) \wedge (\neg\mathbf{j}_3 \vee [\alpha](\langle\alpha\rangle\langle\alpha\rangle[(\alpha(\iota_2(\alpha,\alpha)))^{-1}](\neg\mathbf{j}_2, [\iota_1^{-1}]\neg\mathbf{j}_1))).$$

Negating, translating into first-order logic, and simplifying, bearing in mind the fixed relations used for the interpretation of the ι's, we obtain as local first-order equivalent:

$$\exists y(R_\alpha xy \wedge \forall z(R_\alpha^2 yz \to \exists u(R_\alpha xu \wedge R_\alpha ux \wedge R_\alpha uz)).$$

$\lhd$

Example 4.4.2 Let $\mathbf{1}$ be a unary and $\mathbf{2}$ a binary modal term. Consider the formula

$$\varphi_2 = [\mathbf{2}](\neg[\mathbf{1}](\neg[\mathbf{1}]p \vee p), p \wedge [\mathbf{1}]\bot).$$

This formula has local first-order correspondent $\forall u_1 \forall u_2 (R_{\mathbf{2}} x u_1 u_2 \to R_{\mathbf{1}} u_1 u_2 \wedge \forall z(\neg R_{\mathbf{1}} u_2 z))$. In fact, it is locally frame equivalent to the inductive formula $[\mathbf{2}](\neg[\mathbf{1}]p, p \wedge [\mathbf{1}]\bot)$. Here is an attempt to execute SQEMA on it:

Pase 1 Preprocessing yields

$$\langle \mathbf{2} \rangle([\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p), \neg p \vee \langle \mathbf{1} \rangle \top).$$

Phase 2 The only initial system is

$$\| \mathbf{i} \Rightarrow \langle \mathbf{2} \rangle([\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p), \neg p \vee \langle \mathbf{1} \rangle \top)\ .$$

Applying the $\Diamond$-rule yields

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \mathbf{2} \rangle(\mathbf{j}_1, \mathbf{j}_2) \\ \mathbf{j}_1 \Rightarrow [\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p) \\ \mathbf{j}_2 \Rightarrow \neg p \vee \langle \mathbf{1} \rangle \top \end{array} \right. .$$

We can now apply the $\Box$-rule to the second equation giving a system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \mathbf{2} \rangle(\mathbf{j}_1, \mathbf{j}_2) \\ \langle \mathbf{1}^{-1} \rangle \mathbf{j}_1 \Rightarrow \langle \mathbf{1} \rangle \neg p \vee p \\ \mathbf{j}_2 \Rightarrow \neg p \vee \langle \mathbf{1} \rangle \top \end{array} \right.$$

but it is clear that we cannot separate the positive and negative occurrences of p in the second sequent. The algorithm terminates with failure.

$\lhd$

4.5 Correctness and canonicity

We have to show that the extended algorithm is still correct, and that moreover all formulae on which it succeeds exhibit the desired type of persistence. We will not be detained long by the proof of correctness — proposition 2.4.5 extends in a straight-forward way to the generalized transformation rules, and then the proof of the next theorem is verbatim the same as that of theorem 2.4.6.

Theorem 4.5.1 (Correctness of Polyadic SQEMA) *Let τ be a (polyadic) modal similarity type. If SQEMA succeeds on an input formula $\varphi \in \mathcal{L}_\tau$ or $\varphi \in \mathcal{L}_{r(\tau)}$, then the first-order formula returned is a local frame correspondent of φ.*

The proof of the corresponding analogue of theorem 2.5.23 however, involves some non-trivial complications, relating to the proof of a suitable analogue of Esakia's lemma for diamonds $\langle \alpha \rangle$ for arbitrary $\alpha \in \mathrm{MT}_{r(\tau)}$.

4.5.1 The topology of polyadic descriptive frames

The notion of a general frame extends to the polyadic case in the expected way. To be precise, a *general τ-frame* is a structure $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W})$ where $(W, \{R_\alpha\}_{\alpha\in\tau})$ is a τ-frame, and $\mathbb{W}$ is a Boolean algebra of *admissible subsets* of W, closed under the modal operators $\langle\alpha\rangle$, $\alpha \in \tau$, defined as

$$\langle\alpha\rangle(X_1, \ldots, X_{\rho(\alpha)}) = \{y \in W \mid R_\alpha(y, x_1, \ldots, x_{\rho(\alpha)}) \text{ for some } x_1 \in X_1, \ldots, x_{\rho(\alpha)} \in X_{\rho(\alpha)}\}.$$

Clearly, $\mathbb{W}$ is also closed under the dual operators $[\alpha]$, given by

$$[\alpha](X_1, \ldots, X_{\rho(\alpha)}) = \{y \in W \mid R_\alpha(y, x_1, \ldots, x_{\rho(\alpha)}) \text{ implies } x_1 \in X_1 \text{ or } \ldots \text{ or } x_{\rho(\alpha)} \in X_{\rho(\alpha)}\}.$$

Note that we only demand closure under $\langle\alpha\rangle$ for $\alpha \in \tau$, i.e. for *basic* modal terms α. However, a straightforward inductive argument suffices to see that this is, in fact, enough to guarantee closure under $\langle\alpha\rangle$ for *all* $\alpha \in \mathrm{MT}_\tau$. Of course, as illustrated by example 2.5.2, this need not be the case for inverse diamond operators. This, as in the monadic case, precludes a direct adaptation of Ackermann's lemma to the case of languages containing inverse modalities interpreted over arbitrary descriptive frames. In order to prove the desired analogue, we have to undertake a closer analysis of the topology of $\mathcal{L}^n_{r(\tau)}$-formulae when regarded as operators on descriptive τ-frames. This is what we do in the rest of this section.

A *reversive general τ-frame*, also called a *general $r(\tau)$-frame*, is a general τ-frame $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W})$ where the algebra of admissible sets $\mathbb{W}$ is closed under $\langle\alpha\rangle$, for all $\alpha \in \mathrm{MT}_{\tau(r)}$.

The notions of *differentiatedness* and *compactness* are the same for polyadic general frames as for monadic general frames, but the definition of tightness has to be extended. A relation R_α in $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W})$ is *tight in* $\mathfrak{F}$ if the following condition holds: for any $x, x_1, \ldots, x_n \in W$,

$$R_\alpha x, x_1, \ldots, x_n \text{ iff } \forall X_1, \ldots, X_n \in \mathbb{W}(x_1 \in X_1, \ldots, x_n \in X_n \Rightarrow x \in \langle\alpha\rangle(X_1, \ldots, X_n)).$$

Equivalently, R_α is tight if for all $x, x_1, x_2, \ldots, x_n \in W$,

$$R_\alpha x, x_1, \ldots, x_n \text{ iff } x \in \bigcap\{\langle\alpha\rangle(X_1, \ldots, X_n) \mid X_1, \ldots, X_n \in \mathbb{W} \ \& \ x_1 \in X_1, \ldots, x_n \in X_n\}.$$

Definition 4.5.2 A general τ-frame or $r(\tau)$-frame $\mathfrak{F}$ is *tight* if the relation R_α is tight in $\mathfrak{F}$ for every *basic* modal term $\alpha \in \tau$. A general τ-frame or $r(\tau)$-frame is *descriptive* if it differentiated, tight, and compact.

Given a descriptive τ-frame $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W})$, a subset of W^n, $n \in \mathbb{N}^+$, is said to be *closed* if it is closed in the respective product (or Tychonoff) topology on W^n. Here are some facts about this topology that we will need. Each item is either well-known from topology (see e.g. [Wil04]) or easy to see.

- We can take as a base for the product topology on W^n all sets of the from $A_1 \times \cdots \times A_n$ where $A_1, \ldots, A_n \in \mathbb{W}$.

- In particular, for any $C_1, \ldots, C_n \in \mathbf{C}(\mathbb{W})$ it will be the case that $C_1 \times \cdots \times C_n$ is closed in the product topology on W^n.
- Any set $C \subseteq W^n$ which is closed in W^n can be written as $C = \bigcap_{i \in I} -(A_{1_i} \times \cdots \times A_{n_i})$ for some index set I and $A_{j_i} \in \mathbb{W}$, $1 \leq j \leq n$, $i \in I$ and as $C = \bigcap_{i \in I} -(-A_{1_i} \times \cdots \times -A_{n_i})$ for some index set I and $A_{j_i} \in \mathbb{W}$, $1 \leq j \leq n$, $i \in I$.
- If R_α, $\alpha \in \mathrm{MT}_{r(\tau)}$ is tight, then $R_\alpha(x) = \{(y_1, \ldots y_n) \in W^n \mid R_\alpha x y_1 \ldots yn\}$ is closed in the product topology on W^n, i.e. R_α is *point-closed* (cf. also [GV06], lemma 50).
- If By Tychonoff's theorem, if $T(\mathfrak{F})$ is compact, then product topology on W^n will be compact.

The notions of *augmented valuations, augmented models* and *augmented satisfiability* (definition 2.5.3) and *ad-persistence* (definition 2.5.4) are extended to descriptive τ-frames and descriptive $r(\tau)$-frames in the obvious way. We will write $\varphi \equiv^{ad\,\tau}_{\text{trans}} \psi$ if φ and ψ are ad-transformation equivalent on descriptive τ-frames, and $\varphi \equiv^{ad\,r(\tau)}_{\text{trans}} \psi$ to indicate that they are ad-transformation equivalent on descriptive $r(\tau)$-frames.

4.5.2 $\mathcal{L}^n_{r(\tau)}$-formulae as operators on descriptive τ-frame

The notions of closed (open) formulae and closed (open) operators (definition 2.5.9) generalize to $\mathcal{L}^n_{r(\tau)}$-formulae interpreted over descriptive τ-frames and descriptive $r(\tau)$-frames. The proofs of the results in this section are essentially the same as those in [CGV06b].

Lemma 4.5.3 *The formulae $\langle\alpha\rangle(p_1, \ldots, p_n)$ and $[\alpha](p_1, \ldots, p_n)$ are, respectively, open and closed operators on arbitrary τ-frames, for any $\alpha \in \mathrm{MT}_\tau$ with $\rho(\alpha) = n$. Similarly, for any $\alpha \in \mathrm{MT}_{r(\tau)}$ with $\rho(\alpha) = n$, the formulae $\langle\alpha\rangle(p_1, \ldots, p_n)$ and $[\alpha](p_1, \ldots, p_n)$ are, respectively, open and closed operators on arbitrary $r(\tau)$-frames.*

Proof. We prove the case for $\alpha \in \mathrm{MT}_\tau$, the other is similar. Let $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \tau})$ be any τ-frame and $\alpha \in \mathrm{MT}_\tau$. Let $A_1, \ldots, A_n \in \mathbf{C}(\mathfrak{F})$ be closed sets in $\mathfrak{F}$. Then there are families of admissible sets $\{A^1_i\}_{i \in I_1}, \ldots, \{A^n_i\}_{i \in I_n}$ such that $A_1 = \bigcap_{i \in I_1} \{A^1_i\}$, …, $A_n = \bigcap_{i \in I_n} \{A^n_i\}$. Hence $[\alpha](A_1, \ldots, A_n) = [\alpha](\bigcap_{i \in I_1} A^1_i, \ldots, \bigcap_{i \in I_n} A^n_i)$. But

$$[\alpha](\bigcap_{i \in I_1} A^1_i, \ldots, \bigcap_{i \in I_n} A^n_i) = \bigcap_{i \in I_1} \cdots \bigcap_{i \in I_n} [\alpha](A^1_i, \ldots, A^n_i).$$

The latter is an intersection of admissible sets, and hence closed. The case for $\langle\alpha\rangle$ is similar, using the fact that diamonds distribute over arbitrary unions. QED

The following polyadic version of lemma 2.5.10 was proven in [GV06].

Lemma 4.5.4 (Esakia's Lemma for Diamonds, [GV06]) *Let $\mathfrak{F}$ be a descriptive τ-frame and $\alpha \in \mathrm{MT}_{r(\tau)}$ and n-ary modal term, such that either,*

1. *$\alpha \in \mathrm{MT}_\tau$, or*

2. *$\mathfrak{F}$ is reversive,*

then, for any downward directed family $\{X_{1_i} \times \cdots \times X_{n_i} \mid i \in I\}$ of nonempty closed subsets of W^n, and any n-ary $\alpha \in \mathrm{MT}_\tau$, it is the case that

$$\bigcap_{i\in I}\langle\alpha\rangle(X_{1_i}, \ldots, X_{n_i}) = \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \ldots, \bigcap_{i\in I} X_{n_i}).$$

Using the fact that

Corollary 4.5.5 *For any n-ary $\alpha \in \mathrm{MT}_\tau$, (respectively, $\alpha \in \mathrm{MT}_{r(\tau)}$), $\langle\alpha\rangle(p_1, \ldots, p_n)$ is both a closed and open operator on descriptive τ-frames (respectively, descriptive $r(\tau)$-frames).*

By the duality of $\langle\alpha\rangle$ and $[\alpha]$ we have:

Corollary 4.5.6 *For any n-ary $\alpha \in \mathrm{MT}_\tau$, (respectively, $\alpha \in \mathrm{MT}_{r(\tau)}$), $[\alpha](p_1, \ldots, p_n)$ is both a closed and open operator on descriptive τ-frames (respectively, descriptive $r(\tau)$-frames).*

We will need results analogous to lemma 4.5.4 and corollaries 4.5.5 and 4.5.6 for the inverse diamonds from $\mathrm{MT}_{r(\tau)}$ in descriptive but not necessarily reversive general frames.

Recall that the definition of a descriptive τ-frame only requires tightness for relations R_α for *basic* modal terms $\alpha \in \tau$. The next lemma shows that this is enough to guarantee tightness of R_α for *all* $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$. It also simultaneously lifts corollary 4.5.5 to $\langle\alpha\rangle$ for arbitrary $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$.

Lemma 4.5.7 *For any n-ary $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$,*

1. *$\langle\alpha\rangle(p_1, \ldots, p_n)$ is a closed operator on descriptive τ-frames, and*

2. *R_α is tight in any descriptive τ-frame.*

Proof. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a (not necessarily reversive) descriptive τ-frame. The proof proceeds by induction on $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$. The base case is for basic modal terms $\alpha \in \tau$, and holds by corollary 4.5.5 and the definition of descriptive τ-frames.

Now suppose that $\alpha, \beta_1, \ldots, \beta_n \in \mathrm{MT}_\tau$ with $\rho(\alpha) = n$, $\rho(\beta_1) = m_1, \ldots, \rho(\beta_n) = m_n$, such that $\langle\alpha\rangle(p_1, \ldots, p_n), \ldots, \langle\beta_1\rangle(p_1, \ldots, p_{m_1}), \ldots, \langle\beta_n\rangle(p_1, \ldots, p_{m_n})$ are closed operators on descriptive frames and such that $R_\alpha, R_{\beta_1}, \ldots, R_{\beta_n}$ are tight in $\mathfrak{F}$. It is trivial to see that $\langle\alpha(\beta_1, \ldots, \beta_n)\rangle(p_{1_1}, \ldots, p_{m_1}, \ldots, p_{1_n}, \ldots, p_{m_n})$ is a closed operator.

We have to show that $R_{\alpha(\beta_1,\ldots,\beta_n)}$ is tight in $\mathfrak{F}$. To keep the notation manageable we treat only binary terms, i.e. suppose that $n = m_1 = m_2 = 2$. We have to show that $R_{\alpha(\beta_1,\beta_2)}$ is tight in $\mathfrak{F}$. To that end, suppose that $\neg R_{\alpha(\beta_1,\beta_2)} y_0 u_1 u_2 v_1 v_2$. It is sufficient to show that $y_0 \notin \bigcap\{\langle\alpha(\beta_1, \beta_2)\rangle(U_1, U_2, V_1, V_2) \mid U_1, U_2, V_1, V_2 \in \mathbb{W} \ \&\ u_1 \in U_1, u_2 \in U_2, v_1 \in V_1, v_2 \in V_2\}$. For every pair $z_1, z_2 \in W$ such that $R_\alpha y_0 z_1 z_2$ it is the case that $\neg R_{\beta_1} z_1 u_1 u_2$ or $\neg R_{\beta_2} z_2 v_1 v_2$. Hence, by the tightness of R_{β_1} and R_{β_2}, for every pair $z_1, z_2 \in W$ such that $R_\alpha y_0 z_1 z_2$ there exist $U_1, U_2 \in \mathbb{W}$ such that $u_1 \in U_1$, $u_2 \in U_2$, and $z_1 \notin \langle\beta_1\rangle(U_1, U_2)$, or there exist $V_1, V_2 \in \mathbb{W}$ such that $v_1 \in V_1$, $v_2 \in V_2$, and $z_2 \notin \langle\beta_2\rangle(V_1, V_2)$. Hence $R_\alpha(y_0) \cap \bigcap\{\langle\beta_1\rangle(U_1, U_2) \times \langle\beta_2\rangle(V_1, V_2) \mid U_1, U_2, V_1, V_2 \in \mathbb{W} \ \&\ u_1 \in U_1, u_2 \in U_2, v_1 \in V_1, v_2 \in V_2\} = \emptyset$. Now $R_\alpha(y_0)$ is closed by the tightness of R_α, and by the inductive hypothesis $\langle\beta_1\rangle$ and $\langle\beta_2\rangle$ are closed operators. Hence we have a family of closed sets with empty intersection. By appealing to compactness and the monotonicity of $\langle\beta_1\rangle$ and $\langle\beta_2\rangle$ we conclude that there exist sets $U'_1, U'_2, V'_1, V'_2 \in \mathbb{W}$ such that $u_1 \in U'_1, u_2 \in U'_2, v_1 \in V'_1, v_2 \in V'_2$ and $R_\alpha(y_0) \cap \langle\beta_1\rangle(U'_1, U'_2) \times \langle\beta_2\rangle(V'_1, V'_2) = \emptyset$. It follows that $y_0 \notin \bigcap\{\langle\alpha(\beta_1,\beta_2)\rangle(U_1, U_2, V_1, V_2) \mid U_1, U_2, V_1, V_2 \in \mathbb{W} \ \&\ u_1 \in U_1, u_2 \in U_2, v_1 \in V_1, v_2 \in V_2\}$ as desired. This concludes the inductive step for compositions of modal terms.

Instead of a inductive step for inverses, we do an inductive step for permutations. Let σ be any permutation of $\{0, 1, \ldots, n\}$.

First we show that R^σ_α is tight in $\mathfrak{F}$. To that end assume that $\neg R^\sigma_\alpha y_0, y_1, \ldots y_n$. We have to show that $y_0 \notin \bigcap\{\langle\alpha^\sigma\rangle(X_1, \ldots, X_n) \mid X_1, \ldots, X_n \in \mathbb{W} \ \&\ y_1 \in X_1, \ldots, y_n \in X_n\}$. We have that $\neg R_\alpha y_{\sigma(0)}, y_{\sigma(1)}, \ldots, y_{\sigma(n)}$, and hence, by the tightness of R_α, that $y_\sigma(0) \notin \bigcap\{\langle\alpha\rangle(X_1, \ldots, X_n) \mid X_1, \ldots, X_n \in \mathbb{W} \ \&\ y_{\sigma(1)} \in X_1, \ldots, y_{\sigma(n)} \in X_n\}$. Hence there exist $U_1, \ldots, U_n \in \mathbb{W}$ such that $y_{\sigma(1)} \in U_1, \ldots, y_{\sigma(n)} \in U_n$ but such that $y_{\sigma(0)} \notin \langle\alpha\rangle(U_1, \ldots, U_n)$, i.e. $\{y_{\sigma(0)}\} \cap \langle\alpha\rangle(U_1, \ldots, U_n) = \emptyset$. But $\mathfrak{F}$ is differentiated, hence $\{y_{\sigma(0)}\} = \bigcap\{A \in \mathbb{W} \mid y_{\sigma(0)} \in A\}$. Furthermore, since $\langle\alpha\rangle$ is a closed operator, $\langle\alpha\rangle(U_1, \ldots, U_n)$ is a closed set. Hence, by compactness, it follows that there exists a single admissible set $U_0 \in \mathbb{W}$ such that $y_{\sigma(0)} \in U_0$ and $U_0 \cap \langle\alpha\rangle(U_1, \ldots, U_n) = \emptyset$. Hence $U_{\overline{\sigma}(0)} \cap \langle\alpha^\sigma\rangle(U_{\overline{\sigma}(1)}, \ldots, U_{\overline{\sigma}(n)}) = \emptyset$. But $y_i \in U_{\overline{\sigma}(i)}$, for all $0 \leq i \leq n$. Hence $y_0 \notin \bigcap\{\langle\alpha^\sigma\rangle(X_1, \ldots, X_n) \mid X_1, \ldots, X_n \in \mathbb{W} \ \&\ y_1 \in X_1, \ldots, y_n \in X_n\}$.

Next we show that $\langle\alpha^\sigma\rangle(p_1, \ldots, p_n)$ is a closed operator. To that end let $(A_1, \ldots, A_n)$ be a tuple of closed sets in $T(\mathfrak{F})$. We have to show that $\langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$ is a closed set in $T(\mathfrak{F})$. We will split the proof into two cases, according to whether $\sigma(0) = 0$ or $\sigma(0) \neq 0$.

Case 1: Suppose $\sigma(0) = 0$. We claim that

$$\langle\alpha^\sigma\rangle(A_1, \ldots, A_n) = \langle\alpha\rangle(A_{\sigma(1)}, \ldots, A_{\sigma(n)}).$$

The closedness of $\langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$ then follows from the fact that $\langle\alpha\rangle$ is a closed operator. Indeed, if $x_0 \in \langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$ then there exist $x_1 \in A_1, \ldots, x_n \in A_n$ such that $R^\sigma_\alpha x_0 x_1 \ldots x_n$. Then $R_\alpha x_{\sigma(0)} x_{\sigma(1)} \ldots x_{\sigma(n)}$, i.e. $R_\alpha x_0 x_{\sigma(1)} \ldots x_{\sigma(n)}$. Hence $x_0 \in \langle\alpha\rangle(A_{\sigma(1)}, \ldots, A_{\sigma(n)})$.

Conversely, if $x_0 \in \langle\alpha\rangle(A_{\sigma(1)}, \ldots, A_{\sigma(n)})$ then there exist $x_1 \in A_{\sigma(1)}, \ldots, x_n \in A_{\sigma(n)}$ (i.e. $x_{\overline{\sigma}(1)} \in A_1, \ldots, x_{\overline{\sigma}(n)} \in A_n$) such that $R_\alpha x_0 x_1 \ldots x_n$. Then $R^\sigma_\alpha x_{\overline{\sigma}(0)} x_{\overline{\sigma}(1)} \ldots x_{\overline{\sigma}(n)}$, i.e. $R^\sigma_\alpha x_0 x_{\overline{\sigma}(1)} \ldots x_{\overline{\sigma}(n)}$. Hence $x_0 \in \langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$. Thus the proof of case 1 is concluded.

Case 2:[2] Suppose $\sigma(0) = k \neq 0$. Hence also $\overline{\sigma}(0) \neq 0$, say $\overline{\sigma}(0) = j$. To show that $\langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$ is a closed set in $T(\mathfrak{F})$ it is enough to prove the equality

$$\langle\alpha^\sigma\rangle(A_1, \ldots, A_n) = \bigcap\{B \in \mathbb{W} \mid \langle\alpha^\sigma\rangle(A_1, \ldots, A_n) \subseteq B\}. \tag{4.1}$$

The left-to-right inclusion is trivial. For the sake of the right-to-left inclusion suppose that $y_0 \notin \langle\alpha^\sigma\rangle(A_1, \ldots, A_n)$. Hence we have

$$R^\sigma_\alpha(y_0) \cap (A_1 \times \ldots \times A_n) = \emptyset. \tag{4.2}$$

[2]The proof of this case generalizes, and closely follows, that of theorem 72 in [GV06], where the current theorem is proved for the more restricted case of modal terms in $MT_{\tau r}$ rather than $MT_{r(\tau)}$.

This means that

$$\forall y_1 \ldots \forall y_n((y_1,\ldots,y_n) \in A_1 \times \cdots \times A_n \to (y_1,\ldots,y_n) \notin R^\sigma_\alpha(y_0)) \tag{4.3}$$

i.e.,

$$\forall y_1 \ldots \forall y_n((y_1,\ldots,y_n) \in A_1 \times \cdots \times A_n \to \neg R^\sigma_\alpha y_0 y_1,\ldots,y_n), \tag{4.4}$$

i.e.

$$\forall y_1 \ldots \forall y_n((y_1,\ldots,y_n) \in A_1 \times \cdots \times A_n \to \neg R_\alpha y_{\sigma(0)}, y_{\sigma(1)} \ldots, y_{\sigma(n)}). \tag{4.5}$$

Note that in (4.5) y_0 appears as the $\overline{\sigma}(0)$-th, i.e. j-th argument of R_α. Now by the tightness of R_α, $R_\alpha(y_{\sigma(0)})$ is a closed set, i.e.

$$R_\alpha(y_{\sigma(0)}) = \bigcap\{-(-B_1 \times \ldots \times -B_n) \mid R_\alpha(y_{\sigma(0)}) \subseteq -(-B_1 \times \ldots \times -B_n), B_i \in \mathbb{W}\} \tag{4.6}$$

i.e.,

$$R_\alpha(y_{\sigma(0)}) = \bigcap\{-(-B_1 \times \ldots \times -B_n) \mid y_{\sigma(0)} \in [\alpha](B_1,\ldots,B_n), B_i \in \mathbb{W}\}. \tag{4.7}$$

From (4.7) and (4.5) it follows that for each $\overline{y} = (y_1,\ldots,y_n) \in A_1 \times \cdots \times A_n$ there exist sets $B_1^{\overline{y}},\ldots,B_n^{\overline{y}} \in \mathbb{W}$ such that $y_{\sigma(0)} \in [\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}})$ and $(y_{\sigma(1)},\ldots,y_{\sigma(n)}) \notin -(-B_1^{\overline{y}} \times \ldots \times -B_n^{\overline{y}})$, i.e. $y_{\sigma(0)} \in [\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}})$ and $(y_{\sigma(1)},\ldots,y_{\sigma(n)}) \in (-B_1^{\overline{y}} \times \ldots \times -B_n^{\overline{y}})$. For the rest of the proof fix such $B_1^{\overline{y}},\ldots,B_n^{\overline{y}}$ for each $\overline{y} \in A_1 \times \cdots \times A_n$. Specifically note that for each $\overline{y} \in A_1 \times \cdots \times A_n$ we have $y_0 = y_{\sigma(j)} \notin B_j^{\overline{y}}$. Hence we have

$$A_1 \times \cdots \times A_n \subseteq \bigcup\{(-B^{\overline{y}}_{\overline{\sigma}(1)}) \times \cdots \times [\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}}) \times \cdots \times (-B^{\overline{y}}_{\overline{\sigma}(n)}) \mid \overline{y} \in A_1 \times \cdots \times A_n\} \tag{4.8}$$

where $[\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}})$ is the in the $\sigma(0)$-th,. i.e. k-th, coordinate of the product. Note that $A_1 \times \cdots \times A_n$, being a product of closed sets, is closed in the product topology, and that $\bigcup\{(-B^{\overline{y}}_{\overline{\sigma}(1)}) \times \cdots \times [\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}}) \times \cdots \times (-B^{\overline{y}}_{\overline{\sigma}(n)}) \mid \overline{y} \in A_1 \times \cdots \times A_n\}$ forms an open cover of $A_1 \times \cdots \times A_n$. It follows by compactness that there are finite sets $A_1',\ldots,A_n'$ such that $A_1' \subseteq A_1,\ldots,A_n' \subseteq A_n$ and

$$A_1 \times \cdots \times A_n \subseteq \bigcup\{(-B^{\overline{y}}_{\overline{\sigma}(1)}) \times \cdots \times [\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}}) \times \cdots \times (-B^{\overline{y}}_{\overline{\sigma}(n)}) \mid \overline{y} \in A_1' \times \cdots \times A_n'\} \tag{4.9}$$

By the monotonicity of $\langle\alpha^\sigma\rangle$ and its distributivity over arbitrary unions we have

$$\langle\alpha^\sigma\rangle(A_1,\ldots,A_n) \subseteq \bigcup\{\langle\alpha^\sigma\rangle(-B^{\overline{y}}_{\overline{\sigma}(1)},\ldots,[\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}}),\ldots,-B^{\overline{y}}_{\overline{\sigma}(n)}) \mid \overline{y} \in A_1' \times \cdots \times A_n'\}. \tag{4.10}$$

By the contrapositive of the implication proved in lemma 4.1.2 we have, for each $\overline{y} \in A_1' \times \cdots \times A_n'$,

$$\langle\alpha^\sigma\rangle(-B^{\overline{y}}_{\overline{\sigma}(1)},\ldots,[\alpha](B_1^{\overline{y}},\ldots,B_n^{\overline{y}}),\ldots,-B^{\overline{y}}_{\overline{\sigma}(n)}) \subseteq B_j^{\overline{y}}. \tag{4.11}$$

Hence we have

$$\langle\alpha^\sigma\rangle(A_1,\ldots,A_n) \subseteq \bigcup\{B_j^{\overline{y}} \mid \overline{y} \in A_1' \times \cdots \times A_n'\} = B_0 \tag{4.12}$$

But then $B_0 \in \mathbb{W}$ and $y_0 \notin B_0$, and we are done. QED

By the duality of $\langle\alpha\rangle$ and $[\alpha]$, we obtain:

Corollary 4.5.8 *For any n-ary modal term $\alpha \in \mathrm{MT}_{r(\tau)}$, it is the case that $[\alpha](p_1, \dots, p_n)$ is an open operator on descriptive τ-frames.*

Now, we are ready to prove the version of Esakia's Lemma for inverses of diamonds from $\mathrm{MT}_{r(\tau)}$ on any descriptive τ-frame.

Lemma 4.5.9 (Esakia's Lemma for inverse diamonds from $\mathrm{MT}_{r(\tau)}$) *Let $\alpha \in \mathrm{MT}_{r(\tau)}$ be an n-ary modal term, and $\mathfrak{F}$ any descriptive τ-frame. Then*

$$\langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \bigcap_{i\in I} \langle\alpha\rangle(X_{1_i}, \dots, X_{n_i})$$

whenever $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i\in I}$ is a family of downwards directed sets such that X_{j_i} is closed in $T(\mathfrak{F})$ for all $1 \le j \le n$ and $i \in I$.

Proof. The inclusion from left to right is immediate by the monotonicity of $\langle\alpha\rangle$. For the other direction, suppose that $x_0 \notin \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \dots, \bigcap_{i\in I} X_{n_i})$, i.e.

$$\{x_0\} \cap \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Hence

$$\bigcap_{i\in I} X_{1_i} \cap \langle\alpha^{-1}\rangle(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Now, since $\{x_0\}$ is closed in $T(\mathfrak{F})$, by lemma 4.5.7 we have here a family of closed sets with empty intersection. By compactness, there is a finite subfamily with empty intersection, say

$$X_{1_{i_1}} \cap \cdots \cap X_{1_{i_m}} \cap \langle\alpha^{-1}\rangle(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Furthermore, since $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i\in I}$ is downward directed, then so is every family $\{X_{1_i}\}_{i\in I}$, $\cdots$, $\{X_{n_i}\}_{i\in I}$. Therefore, we can find a $X_1 \in \{X_{1_i}\}_{i\in I}$ such that $X_1 \subseteq X_{1_{i_1}} \cap \cdots \cap X_{1_{i_m}}$, and hence

$$X_1 \cap \langle\alpha^{-1}\rangle(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Equivalently, it must be the case that

$$\bigcap_{i\in I} X_{2_i} \cap \langle\alpha^{-2}\rangle(X_1, \{x_0\}, \bigcap_{i\in I} X_{3_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

In the same way as above, we find a $X_2 \in \{X_{2_i}\}_{i\in I}$ such that

$$X_2 \cap \langle\alpha^{-2}\rangle(X_1, \{x_0\}, \bigcap_{i\in I} X_{3_i}, \dots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Proceeding likewise, we find $X_3 \in \{X_{3_i}\}_{i \in I}$, ..., $X_n \in \{X_{n_i}\}_{i \in I}$ such that

$$\{x_0\} \cap \langle\alpha\rangle(X_1, \ldots, X_j, \ldots, X_n) = \emptyset.$$

Therefore,

$$x_0 \notin \bigcap_{i_1,\ldots,i_n \in I} \langle\alpha\rangle(X_{1_{i_1}}, \ldots, X_{n_{i_n}}).$$

The result follows once we note that, by the downward directedness of $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i \in I}$,

$$\bigcap_{i_1,\ldots,i_n \in I} \langle\alpha^{-j}\rangle(X_{1_{i_1}}, \ldots, X_{n_{i_n}}) = \bigcap_{i \in I} \langle\alpha^{-j}\rangle(X_{1_i}, \ldots, X_{n_i}).$$

QED

The the next definition generalizes the notions of syntactic openness and closedness to polyadic languages, and also introduces a weaker version of these notions appropriate for reversive τ-frames.

Definition 4.5.10 By an *inverse τ-diamond (τ-box)* we mean a diamond (box) $\langle\alpha\rangle$ ($[\alpha]$) with $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)} - \mathrm{MT}_\tau$.

1. A formula $\varphi \in \mathcal{L}^n_{r(\tau)}$ is *syntactically closed* if all occurrences of nominals and inverse τ-diamonds in φ are positive, and all occurrences of inverse τ-boxes in φ are negative; if the formula is in negation normal form, the latter simply means that it contains no inverse τ-boxes. Likewise, φ is *syntactically open* if all occurrences of nominals and inverse τ-diamonds in φ are negative, and all occurrences of inverse τ-boxes in φ are positive. Clearly, $\neg$ maps syntactically open formulae to syntactically closed formulae, and vice versa.

2. An $\mathcal{L}^n_{r(\tau)}$-formula is called *nominal-negative* (respectively, *nominal-positive*) if all occurrences of nominals in it are negative (respectively, positive), i.e., within the scope of an odd (respectively, even) number of negations. Clearly, negation maps nominal-positive formulae to nominal-negative ones, and vice versa.

Lemma 4.5.11

1. *On any reversive descriptive τ-frame every nominal-negative $\mathcal{L}^n_{r(\tau)}$-formula is an open formula and every nominal-positive $\mathcal{L}^n_{r(\tau)}$-formula is a closed formula.*

2. *On any descriptive τ-frame every syntactically closed $\mathcal{L}^n_{r(\tau)}$-formula is a closed formula and every syntactically open $\mathcal{L}^n_{r(\tau)}$-formula is an open formula.*

Proof. In both cases, by straightforward structural induction on the respective type of formulae, written in negation normal form, using the facts that singletons are closed sets, and that, according to corollaries 4.5.5 and 4.5.6, $\langle\alpha\rangle$ and $[\alpha]$ are both open and closed operators on descriptive frames (respectively, reversive descriptive frames), for any $\alpha \in \mathrm{MT}_\tau$, (respectively, any $\alpha \in \mathrm{MT}_{r(\tau)}$). In the second case we also have to appeal to lemma 4.5.7 and corollary 4.5.8. QED

Lemma 4.5.12 *Let the formula $\varphi(q_1, \dots, q_n, p, \mathbf{i}_1, \dots \mathbf{i}_m) \in \mathcal{L}^n_{r(\tau)}$ be positive in p and $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in MT_\tau}, \mathbb{W})$ be a descriptive τ-frame, such that one of the following holds:*

1. *φ is syntactically closed, or*

2. *φ is nominal-positive and $\mathfrak{F}$ is reversive.*

Then φ is a closed operator with respect to p, i.e., for all $Q_1, \dots, Q_n \in \mathbb{W}$, $x_1, \dots, x_m \in W$, if $C \in \mathbf{C}(\mathbb{W})$, then $\varphi(Q_1, \dots, Q_n, C, \{x_1\}, \dots, \{x_m\}) \in \mathbf{C}(\mathbb{W})$.

Proof. By structural induction on φ written in negation normal form. Consider the first case. Then no subformula of φ can be of the form $[\alpha](\gamma_1, \dots, \gamma_n)$, with $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)} - \mathrm{MT}_\tau$. The inductive step for $\langle\alpha\rangle$, $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$, follows by lemma 4.5.7. The inductive step for $[\alpha]$ $\alpha \in \mathrm{MT}_\tau$, follows from lemma 4.5.3.

The proof for the second case is essentially the same, the inductive step for $[\alpha]$, $\alpha \in \mathrm{MT}_{r(\tau)}$, now follows by lemma 4.5.3 and that for $\langle\alpha\rangle$, $\alpha \in \mathrm{MT}_{r(\tau)}$, by corollary 4.5.5. QED

Lemma 4.5.13 (Esakia's lemma: syntactically closed / nominal positive-formulae) *Let $\varphi(q_1, \dots, q_n, p, \mathbf{i}_1, \dots \mathbf{i}_m) \in \mathcal{L}^n_{r(\tau)}$ be positive in p and let $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}_\tau}, \mathbb{W})$ be a descriptive τ-frame, such that one of the following holds:*

1. *φ is syntactically closed, or*

2. *φ is nominal-positive and $\mathfrak{F}$ is reversive.*

Then for all $Q_1, \dots, Q_n \in \mathbb{W}$, $x_1, \dots, x_m \in W$ and any downwards directed family of closed sets $\{C_i \mid i \in I\}$ it is the case that

$$\varphi(Q_1, \dots, Q_n, \bigcap_{i \in I} C_i, \{x_1\}, \dots, \{x_m\}) = \bigcap_{i \in I} \varphi(Q_1, \dots, Q_n, C_i, \{x_1\}, \dots, \{x_m\}).$$

Proof. For brevity we will omit the parameters $Q_1, \dots, Q_n$, $\{x_1\}, \dots, \{x_m\}$ when writing (sub) formulae. Consider the first case. The proof is by induction on φ, written in negation normal form. The base cases when φ is $\perp$, a propositional variable or a nominal are trivial, and the inductive steps for the boolean connectives are the same as in the monadic case, given by lemma 2.5.19.

Suppose φ of the form $\langle\alpha\rangle(\gamma_1, \dots, \gamma_n)$, for $\alpha \in \mathrm{MT}_{r(\tau)}$, where $\gamma_1, \dots, \gamma_n$ are syntactically closed and positive in p. We have to show that

$$\langle\alpha\rangle(\gamma_1(\bigcap_{i \in I} C_i), \dots, \gamma_n(\bigcap_{i \in I} C_i)) = \bigcap_{i \in I} \langle\alpha\rangle(\gamma_1(C_i), \dots, \gamma_n(C_i)).$$

By the inductive hypothesis we have

$$\langle\alpha\rangle(\gamma_1(\bigcap_{i \in I} C_i), \dots, \gamma_n(\bigcap_{i \in I} C_i)) = \langle\alpha\rangle(\bigcap_{i \in I} \gamma_1(C_i), \dots, \bigcap_{i \in I} \gamma_n(C_i))$$

If $\gamma_k(C_i) = \emptyset$ for some $i \in I$ and $1 \leq k \leq n$, then, by the monotonicity of the γ_k in p,

$$\langle\alpha\rangle(\gamma_1(\bigcap_{i\in I} C_i), \ldots, \gamma_n(\bigcap_{i\in I} C_i)) = \emptyset = \bigcap_{i\in I}\langle\alpha\rangle(\gamma_1(C_i), \ldots, \gamma_n(C_i)),$$

so we may assume that $\gamma_k(C_i) \neq \emptyset$ for all $i \in I$ and $1 \leq k \leq n$. Then, by lemma 4.5.12, $\{\gamma_1(C_i) \times \cdots \times \gamma_n(C_i) \mid i \in I\}$ is a family of non-empty closed sets. Moreover, this family is downwards directed. For, consider any finite subset $\{\gamma_1(C_i) \times \cdots \times \gamma_n(C_i)\}_{i=1,2,\ldots,m}$ of $\{\gamma_1(C_i) \times \cdots \times \gamma_n(C_i) \mid i \in I\}$. By the downwards directedness of $\{C_i \mid i \in I\}$, there is a $C \in \{C_i \mid i \in I\}$ such that $C \subseteq \bigcap_{i=1}^m C_i$. But then $\gamma_k(C) \in \{\gamma_k(C_i) \mid i \in I\}$ and $\gamma_k(C) \subseteq \bigcap_{i=1}^m \gamma_k(C_i)$ by the upwards monotonicity of γ_k in p, and hence $\gamma_1(C) \times \cdots \times \gamma_n(C) \subseteq \bigcap\{\gamma_1(C_i) \times \cdots \times \gamma_n(C_i)\}_{i=1,2,\ldots,m}$. Now we may apply lemma 4.5.9 and conclude that

$$\langle\alpha\rangle(\gamma_1(\bigcap_{i\in I} C_i), \ldots, \gamma_n(\bigcap_{i\in I} C_i)) = \bigcap_{i\in I}\langle\alpha\rangle(\gamma_1(C_i), \ldots, \gamma_n(C_i)).$$

Lastly, the inductive step for $\varphi = [\alpha](\gamma_1, \ldots, \gamma_n)$, for $\alpha \in \mathrm{MT}_\tau$ follows by the inductive hypothesis and the fact that $[\alpha]$ distributes over arbitrary intersections of subsets of W.

The proof for the second case, when the formula is nominal-positive and $\mathfrak{F}$ is reversive, is almost the same, except that we also treat an inductive step for $[\alpha]$ for arbitrary $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$. This case follows by the distributivity of $[\alpha]$ over arbitrary unions and lemma 4.5.3. QED

Lemma 4.5.14 (Ackermann's lemma for descriptive frames) *Let $A, B(p) \in \mathcal{L}^n_{r(\tau)}$ such that p does not occur in A and $B(p)$ is negative in p. Then*

1. *$((A \to p) \wedge B(p)) \equiv^{ad\,\tau}_{\mathrm{trans}} B(A/p)$ whenever A is syntactically closed and B is syntactically open, and*

2. *$((A \to p) \wedge B(p)) \equiv^{ad\,r(\tau)}_{\mathrm{trans}} B(A/p)$ whenever A is nominal-positive and B is nominal-negative.*

Proof. The proof is essentially the same as that of lemma 2.5.20, appealing to lemmas 4.5.12 and 4.5.13 where the latter appeals to lemmas 2.5.17 and 2.5.19, respectively. QED

4.5.3 Proving canonicity: the polyadic and reversive Cases

Having obtained a suitable version of Ackermann's lemma (lemma 4.5.14), we are now ready to prove the pesistence/canonicity results we set out to prove at the beginning of this section. For the most part we proceed along the lines of the monadic case. To begin with, a straightforward inductive argument, parallelling that of lemma 2.5.21, establishes the following lemma:

Lemma 4.5.15

1. *During the entire (successful or unsuccessful) execution of SQEMA on any input formula from $\mathcal{L}_\tau$, all antecedents of all non-diamond-link sequents are syntactically closed formulae, while all consequents of all non-diamond-link sequents are syntactically open.*

2. *During the entire (successful or unsuccessful) execution of* SQEMA *on any input formula from* $\mathcal{L}_{r(\tau)}$, *all antecedents of all non-diamond-link sequents are nominal-positive formulae, while all consequents of all non-diamond-link sequents are nominal-negative formulae.*

Lemma 4.5.16 *Let* Sys *be a system obtained during the execution of* SQEMA *on an* $\mathcal{L}_\tau$-*formula, and let* Sys′ *be obtained from* Sys *by the application of* SQEMA*-transformation rules. Then* $\mathsf{Form}(\mathsf{Sys}) \equiv^{ad\,tau}_{\text{trans}} \mathsf{Form}(\mathsf{Sys}')$. *Similarly, if* Sys′ *was obtained from* Sys′ *during the executing of* SQEMA *on an* $\mathcal{L}_{r(\tau)}$-*formula, then* $\mathsf{Form}(\mathsf{Sys}) \equiv^{ad\,r(tau)}_{\text{trans}} \mathsf{Form}(\mathsf{Sys}')$.

Proof. The proof is analogous to that of lemma 2.5.22, appealing to lemmas 4.5.14 and 4.5.15, where the latter appeals to lemmas 2.5.20 and 2.5.21, respectively. QED

Theorem 4.5.17

1. *If* SQEMA *succeeds on a* $\mathcal{L}_\tau$-*formula* φ, *then* φ *is locally persistent with respect to the class of all descriptive* τ-*frames.*

2. *If* SQEMA *succeeds on a* $\mathcal{L}_{r(\tau)}$-*formula* φ, *then* φ *is locally persistent with respect to the class of all reversive descriptive* τ-*frames.*

Proof. The proof is completely analogous to that of theorem 2.5.23, appealing to lemma 4.5.16 where the latter appeals to lemma 2.5.22. QED

4.6 Completeness for polyadic inductive formulae

In this section we show that SQEMA succeeds on all polyadic inductive formulae, obtaining as corollaries the local first-order definability and canonicity of these formulae.

Definition 4.6.1 Call a system of SQEMA-equations an *inductive system*, if it has the form

$$\left\| \begin{array}{l} \mathbf{i}_1 \Rightarrow [\beta_1](p_1, N_{1_1}, \ldots, N_{1_m}) \\ \vdots \\ \mathbf{i}_n \Rightarrow [\beta_n](p_n, N_{n_1}, \ldots, N_{n_m}) \\ \mathbf{j}_1 \Rightarrow Neg_1 \\ \vdots \\ \mathbf{j}_k \Rightarrow Neg_k \end{array} \right. ,$$

where

(IS1) either n or k, but obviously not both, may possibly be 0,

(IS2) each $[\beta_i](p_i, N_{i_1}, \ldots, N_{i_m})$ is a headed box with head p_i such that the dependency digraph of this set of boxes is acyclic,

(IS3) every propositional variable occurring in the system occurs at least once as the head of some $[\beta_i](p_i, N_{i_1}, \ldots, N_{i_m})$, and

(IS4) each Neg_i is negative in all occurring propositional variables.

For technical convenience we assume that in inductive systems all heads of boxes occur as the first arguments. Nothing essential in the ensuing changes if we drop this assumption, as the reader can easily verify.

Lemma 4.6.2 *Any inductive system may be transformed into a pure system by application of* SQEMA*-transformation rules.*

Proof. We proceed by induction on the number n of sequents of the form

$$\mathbf{i}_i \Rightarrow [\beta_i](p_i, N_{i_1}, \ldots, N_{i_m})$$

occurring in the system. In any inductive system, every occurring variable must have at least one occurrence as the head of a headed box, so if $n = 0$ the system must be pure.

Assume $n > 1$. Assume further, w.l.o.g., that the variable q is minimal with respect to some fixed linear extension of the partial order induced by the dependency digraph. We can then apply the $\Box$-rule to every sequent $\mathbf{i}_i \Rightarrow [\beta_i](p_i, N_{i_1}, \ldots, N_{i_m})$ which has $p_i = q$, replacing it in the system with $\langle\beta_i^{-1}\rangle(\mathbf{i}_i, \neg N_{i_1}, \ldots, \neg N_{i_m}) \Rightarrow p_i$, where $\langle\beta_i^{-1}\rangle(\mathbf{i}_i, \neg N_{i_1}, \ldots, \neg N_{i_m})$ is a pure formula, by the minimality of $q = p_i$. In this way all positive occurrences of q are separated. Thus the Ackermann-rule now becomes applicable to the system, i.e., we may substitute $\bigvee\{\langle\beta_i^{-1}\rangle(\mathbf{i}_i, \neg N_{i_1}, \ldots, \neg N_{i_m}) \mid p_i = q\}$ for all negative occurrences of q, and remove the sequents $\langle\beta_i^{-1}\rangle(\mathbf{i}_i, \neg N_{i_1}, \ldots, \neg N_{i_m}) \Rightarrow p_i$, $p_i = q$. It is not difficult to see that the system obtained in this way is still an inductive system, now containing at most $n-1$ sequents of the form $\mathbf{i}_j \Rightarrow [\beta_j](p_j, N_{j_1}, \ldots, N_{j_m})$. Finally, we appeal to the inductive hypothesis to conclude that all remaining variables can be eliminated by the application of SQEMA-transformation rules. QED

Theorem 4.6.3 *SQEMA succeeds on all conjunctions of polyadic inductive formulae.*

PROOF: We simply note that, when SQEMA is run on a conjunction of inductive formulae, each initial system of equations (Phase 2.1) is of the type $\|\mathbf{i} \Rightarrow \langle\alpha\rangle(B_1, \ldots, B_n)$ with $\langle\alpha\rangle(B_1, \ldots, B_n)$ the negation of an inductive formula, which, after application of the $\Diamond$-rule, becomes an inductive system. Now, we appeal to lemma 4.6.2. $\Box$

Corollary 4.6.4 *All inductive formulae are locally first-order definable and d-persistent.*

Chapter 5

Hybrid Languages

In this chapter we consider algorithmic correspondence and completeness theory for hybrid languages. Specifically, we adapt SQEMA to deal appropriately with input formulae from the languages $\mathcal{L}^n$, $\mathcal{L}^n_r$, $\mathcal{L}^{n,u}$, $\mathcal{L}^{n,u}_r$, $\mathcal{L}^{n,@}$, and $\mathcal{L}^{n,@}_r$. The rules and strategies that are needed to deal with the additional operators of these languages are usually not surprising. Most of our work will therefore be concerned with showing that the algorithm guarantees suitable types of persistence to the input formulae which it reduces. In the course of this chapter we obtain correspondence and completeness results for several new syntactically specified classes of hybrid formulae.

5.1 The languages $\mathcal{L}^n$, $\mathcal{L}^n_r$ and di-persistence

In this section we consider basic modal and tense languages with the addition of nominals. It is clear that the basic SQEMA-algorithm can be applied to input formulae from these languages without requiring any adaptation. Moreover, the correctness of the algorithm on Kripke frames is unaffected by such an enlargement of its domain of application. However, if we want a concomitant completeness result we will have to prove that all reducible formulae are persistent with respect to some suitable class of general frames. One such class is the class of discrete general frames: The rules of deduction for hybrid logics are designed in such a way that they alow one to build discrete canonical general frames, and hence di-persistence may be seen as a suitable canonicity-notion for hybrid formulae. The following result is well-known, see [GPT87], [GG93] and [BT99].

Theorem 5.1.1 ([GPT87], [GG93], [BT99]) *For any set of $\mathcal{L}^n$-formulae Σ (respectively $\mathcal{L}^{n,u}$, respectively $\mathcal{L}^{n,@}$-formulae) the logic $\mathbf{K}^n \oplus \Sigma$ (respectively $\mathbf{K}^{n,u} \oplus \Sigma$, respectively $\mathbf{K}^{n,@} \oplus \Sigma$) is strongly sound and complete with respect to its class of discrete general frames. This also holds for the reversive versions of the respective languages and logics.*

We would like to show that SQEMA preserves equivalence on discrete general frames. To that end we adapt the notion of transformation equivalence to (reversive) discrete frames:

Definition 5.1.2 Formulae $\varphi, \psi \in \mathcal{L}_r^n$ are *transformation equivalent on discrete frames* (or *di-equivalent*, for short) if, for every model $\mathcal{M} = (\mathfrak{F}, V)$ based on a discrete frame $\mathfrak{F}$, such that $\mathcal{M} \Vdash \varphi$, there exists a $(\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi), \mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi))$-related model $\mathcal{M} = (\mathfrak{F}, V')$ based on $\mathfrak{F}$, such that $\mathcal{M}' \Vdash \psi$, and vice versa. We will write $\varphi \equiv_{\text{trans}}^{di} \varphi$ if φ and ψ are transformation equivalent on discrete frames. We will write $\varphi \equiv_{\text{trans}}^{rdi} \varphi$ if φ and ψ are transformation equivalent on reversive discrete frames.

Note that, since all singletons are admissible in discrete frames, we can now interpret $\mathcal{L}_r^n$-formulae in the standard way, unlike the case for the basic modal language and d-persistence where we had to make use of augmented valuations (section 2.5.2). (Of course the extension of an $\mathcal{L}_r^n$-formula may still produce an inadmissible set in a non-reversive discrete frame.)

5.1.1 The reversive case — $\mathcal{L}_r^n$

The case for $\mathcal{L}_r^n$ is unproblematic, and is facilitated by the following version of Ackermann's lemma. The proof is the same as that of lemma 2.1.4 for the basic modal language, using the fact that the extensions of all $\mathcal{L}_r^n$-formulae are admissible in reversive discrete frames.

Lemma 5.1.3 (Ackermann's lemma for reversive discrete frames) *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a reversive discrete frame, and let $A, B(p) \in \mathcal{L}_r^n$ be such that A does not contain p and $B(p)$ is negative in p. Then, for any model $\mathcal{M}$ based on $\mathfrak{F}$,*

$$\mathcal{M} \Vdash B(A/p)$$

if and only if there exists a model, $\mathcal{M}'$, based on $\mathfrak{F}$ and differing form $\mathcal{M}'$ at most in the valuation of p, such that

$$\mathcal{M}' \Vdash (A \to p) \wedge B(p).$$

Using lemma 5.1.3, all SQEMA-transformation rules are easily shown to preserve transformation equivalence on reversive discrete frames. Combined with the fact that all pure $\mathcal{L}_r^n$-formulae are persistent with respect to reversive discrete frames, this yields the following analogue of theorem 2.5.23.

Theorem 5.1.4 *All formulae from $\mathcal{L}_r^n$ on which SQEMA succeeds are locally persistent with respect to all reversive discrete frames.*

Corollary 5.1.5 *Suppose SQEMA succeeds on every formula in $\Sigma \subseteq \mathcal{L}_r^n$. Then the logics $\mathbf{K}_r^n \oplus \Sigma$, $\mathbf{K}_r^{n,u} \oplus \Sigma$ and $\mathbf{K}_r^{n,@} \oplus \Sigma$ are strongly sound and complete with respect to their classes of Kripke frames.*

5.1.2 The non-reversive case — $\mathcal{L}^n$

The non-reversive case is less direct. Indeed, since the extensions of $\mathcal{L}_r^n$-formulae interpreted on (not necessarily reversive) discrete frames need not be admissible sets in those general frames, we cannot guarantee the soundness of the Ackermann-rule on discrete frames. In fact, it is easy to check that SQEMA succeeds on a formula (provided in [tCMV05]) which axiomatizes an incomplete hybrid logic. It follows, by theorem 5.1.1, that this formula can not be di-persistent, and hence that SQEMA does *not* preserve transformation equivalence on discrete frames. In this subsection we will show how SQEMA can be modified by means of a simple restriction on the application of the Ackermann-rule, to guarantee the local di-persistence of the $\mathcal{L}^n$-formulae it reduces.

Lemma 5.1.6 (Ackermann's lemma for discrete frames) *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a (not necessarily reversive) discrete frame and let $A \in \mathcal{L}^n$ and $B(p) \in \mathcal{L}^n_r$ be such that A does not contain p and $B(p)$ is negative in p. Then, for any model $\mathcal{M}$ based on $\mathfrak{F}$,*

$$\mathcal{M} \Vdash B(A/p)$$

if and only if there exists a model $\mathcal{M}'$, based on $\mathfrak{F}$ and differing form $\mathcal{M}$ at most in the valuation of p, such that

$$\mathcal{M}' \Vdash (A \to p) \wedge B(p).$$

Proof. The bottom-to-top direction follows immediately from the downward monotonicity of $B(p)$ in p and the fact that p does not occur in $B(A/p)$. For the top-to-bottom direction we note that $[\![A]\!]_{\mathcal{M}} \in \mathbb{W}$, since $A \in \mathcal{L}^n_\tau$, hence we can construct $\mathcal{M}'$ from $\mathcal{M}$ simply by letting the valuation of p be equal to $[\![A]\!]_{\mathcal{M}}$. QED

We now modify SQEMA to obtain SQEMA^n by restricting the scope of applications of the Ackermann-rule as follows, in accordance with lemma 5.1.6:

Ackermann-Rule on Discrete Frames: This rule is based on the equivalence given in Ackermann's lemma for discrete frames.

$$\text{The system} \quad \left\Vert \begin{array}{l} A_1 \Rightarrow p \\ \vdots \\ A_n \Rightarrow p \\ B_1(p) \\ \vdots \\ B_m(p) \end{array} \right. \quad \text{is replaced by} \quad \left\Vert \begin{array}{l} B_1[(A_1 \vee \ldots \vee A_n)/p] \\ \vdots \\ B_m[(A_1 \vee \ldots \vee A_n)/p] \end{array} \right. ,$$

where:

1. p does not occur in $A_1, \ldots, A_n$,

2. $A_1, \ldots, A_n \in \mathcal{L}^n$, i.e., these formulae contain no inverse modalities, and

3. each of $B_1, \ldots, B_m$ is negative in p.

Lemma 5.1.7 *Let Sys be a system of SQEMA equations, Sys' be a system obtained from Sys by the application of a transformation rule of SQEMA^n. Then $\mathsf{Form}(\mathsf{Sys}) \equiv^{di}_{\text{trans}} \mathsf{Form}(\mathsf{Sys}')$.*

Proof. It suffices to note that all transformation rules of SQEMA^n maintain transformation equivalence on discrete frames. The case for the Ackermann-rule for discrete frames being justified by lemma 5.1.6. The lemma then follows by appealing to the limited version of transitivity satisfied by transformation equivalence (remark 2.4.3). QED

The proof of the next theorem is directly analogous to that of theorem 2.5.23, appealing to lemma 5.1.7 where the latter appeals to lemma 2.5.8.

Theorem 5.1.8 *Every input formula $\varphi \in \mathcal{L}^n$ on which SQEMA^n succeeds is (locally) di-persistent.*

Corollary 5.1.9 *If SQEMA^n succeeds on all members of a set Σ of $\mathcal{L}^n$-formulae, then the logics $\mathbf{K}^n \oplus \Sigma$, $\mathbf{K}^{n,@} \oplus \Sigma$ and $\mathbf{K}^{n,u} \oplus \Sigma$ are strongly complete with respect to their classes of Kripke frames.*

5.1.3 Syntactic classes

We now demonstrate the scope of SQEMA^n by establishing some completeness results.

Definition 5.1.10 A formula $\varphi \in \mathcal{L}^n$ is *diamond-uniform* if for every propositional variable p occurring in φ, the occurrences of p in φ which are in the scope of a positive diamond or negative box are either all positive, or all negative. Respectively, a formula $\varphi \in \mathcal{L}^n$ is *box-uniform* if, for every propositional variable p occurring in φ, either all occurrences of p in φ in the scope of a negative diamond or positive box are positive, or they are all negative.

Equivalently, a formula $\varphi \in \mathcal{L}^n$ is diamond-uniform if, after transforming φ into negation normal form, for every propositional variable p occurring in φ, either all occurrences of p in φ in the scope of a diamond are positive, or they are all negative. Likewise, the definition of a box-uniform formula in a negation normal form can be simplified.

Clearly, negating a diamond-uniform formula yields a box-uniform formula, and vice versa.

Example 5.1.11 $\Diamond p \to \Box\Diamond p$, and $\Diamond p \to \Diamond\Box p, \Diamond p \to \Diamond\Diamond p$ are diamond-uniform, while $\Box p \to \Diamond p$ and $\Box p \to \Diamond\Box p, \Box p \to \Box\Diamond p$ are not. $\lhd$

Recall the definition of the *very simple Sahlqvist formulae* (definition 1.3.6). The very simple Sahlqvist formulae are probably the best known class of non-pure di-persistent formulae. Note that, in any very simple Sahlqvist formula, every negative occurrence of a variable comes from the antecedent, and is hence not in the scope of any positive diamond. This observation yields the following proposition:

Proposition 5.1.12 *Every very simple Sahlqvist formula is diamond-uniform.*

Diamond uniform-formulae in fact represent a generalization (modulo local equivalence) of the very simple Sahlqvist formulae. Indeed, the formula $\Box p \to \Box\Box p$ is both box and diamond-uniform, but is not a very simple Sahlqvist formula, although it is locally equivalent to the very simple Sahlqvist formula $\Diamond p \to \Diamond\Diamond p$.

Theorem 5.1.13 *SQEMA^n succeeds on all diamond-uniform formulae.*

Proof. We will refer to a system as a *box-uniform system*, if it has the form

$$\left\|\begin{array}{l}\mathbf{i}_1 \Rightarrow \psi_1 \\ \vdots \\ \mathbf{i}_n \Rightarrow \psi_n\end{array}\right. ,$$

where $\psi_1 \wedge \ldots \wedge \psi_n$ is a box-uniform formula, in which, moreover, every occurring disjunction occurs in the scope of a box.

Claim 1 Any propositional variable occurring in a box-uniform system, Sys, can be eliminated from the system by application of transformation rules of SQEMA^n, yielding a system Sys' which is again box-uniform.

Proof of Claim If Sys is box-uniform, then either no positive or no negative occurrence of p in $\mathsf{Form}(\mathsf{Sys})$ is in the scope of any box. Let us consider the first case: It follows that each positive occurrence of p is at most in the scope of diamonds and conjunctions, and hence that the system may be solved for p by the application of the $\Diamond$ and $\wedge$-rules. Observe that applications of the latter rules to box-uniform systems again yield box-uniform systems. When the system is solved for p, all equations containing p positively will be of the form $\mathbf{i}_i \Rightarrow p$. Applying the Ackermann-rule for discrete frames will result in a pure formula being substituted for each negative occurrence of p, thus again yielding a box-uniform system.

In the second case, when no negative occurrence of p in $\mathsf{Form}(\mathsf{Sys})$ is in the scope of any box, we use the polarity switching rule to change the polarity of p and proceed as in the first case. ◀

Note that, when SQEMA^n is run on a diamond-uniform formula φ, the initial system of equation on each disjunctive branch of the execution is a box-uniform system. For, the negation of a diamond-uniform formulae is box-uniform, and, in such a formula, distribution of conjunctions and diamonds over disjunctions ensures that, within the main disjuncts, each disjunction occurs in the scope of a box. A simple inductive argument, appealing to the above claim, now proves the theorem. QED

Corollary 5.1.14 *SQEMA^n succeeds on all very simple Sahlqvist formulae.*

Corollary 5.1.15 *All diamond-uniform formulae are locally di-persistent.*

Example 5.1.16 Consider the Sahlqvist formulae

$$\varphi_1 = \Diamond p \to \Diamond\Box p,\ \varphi_2 = \Diamond p \to \Box\Diamond p,\ \psi_1 = \Box p \to \Diamond\Box p,\ \psi_2 = \Box p \to \Box\Diamond p.$$

1. SQEMA^n succeeds on φ_1 and φ_2 (which are very simple Sahlqvist formulae), but neither on ψ_1 nor on ψ_2.

2. Therefore, both φ_1 and φ_2 are di-persistent. On the other hand, neither ψ_1 nor ψ_2 is di-persistent. That can be seen by checking that both ψ_1 and ψ_2 are valid on the general frame of finite and co-finite subsets of the countably branching tree (where every node has countably many successors)[1], while both fail on the tree itself, taken as a Kripke frame.

$\lhd$

5.2 The languages $\mathcal{L}^n$, $\mathcal{L}^n_r$ and sd-persistence

As we saw in section 5.1, the restrictions that have to be imposed on the Ackermann-rule in order to guarantee di-persistence were quite severe, if fact so severe that the resulting algorithm SQEMA^n will fail even on most Sahlqvist formulae. This is with good reason of course, since most Sahlqvist formulae indeed fail to be di-persistent.

However, it has been shown by Ten Cate in [tC05b] (see also [tCMV05]) that every logic $\mathbf{K}^n \oplus \Sigma$, $\mathbf{K}^{n,@} \oplus \Sigma$ and $\mathbf{K}^{n,u} \oplus \Sigma$ with Σ a set of Sahlqvist formulae (from $\mathcal{L}$) is strongly sound and complete with respect to its class of Kripke frames. In this section we will capture this fact algorithmically via a suitable modification of SQEMA.

It is thus known that hybrid logics axiomatized with pure axioms or with Sahlqvist axioms are complete. However, as [tCMV05] show, we cannot, in general, combine these facts. Specifically, they show that there exists a pure formula $\varphi \in \mathcal{L}^n$ and a Sahlqvist formula $\psi \in \mathcal{L}$ such that $\mathbf{K}^n \oplus \{\varphi, \psi\}$ is incomplete with respect to its class of Kripke frames. Of course, any combination of very simple Sahlqvist formulae and pure formulae axiomatizes a complete hybrid logic, since these formulae are di-persistent.

With the aid of the adaptations of SQEMA that are introduced in this section, we will be able to prove general completeness results for logics axiomatized with $\mathcal{L}^n$-formulae from syntactically specified classes that allow for much more liberal combinations of nominals and propositional variables.

5.2.1 Strongly descriptive frames

The notion of a *strongly descriptive frame* (or sd-frame, for short) is introduced in [tC05b]. Intuitively, a strongly descriptive frame is a descriptive frame that contains 'enough' admissible singletons to allow for the meaningful interpretation of formulae containing nominals. The definition is as follows:

Definition 5.2.1 A general frame $\mathfrak{F} = (W, R, \mathbb{W})$ is *strongly descriptive* if

1. it is descriptive,
2. for all $\emptyset \neq A \in \mathbb{W}$, there is some singleton $\{a\} \in \mathbb{W}$ such that $\{a\} \subseteq A$, and
3. for all $A \in \mathbb{W}$ and singletons $\{a\} \in \mathbb{W}$, if $\{v \in A \mid aRv\} \neq \emptyset$, then there is a singleton $\{b\} \in \mathbb{W}$, such that $b \in A$ and aRb.

[1] This general frame was used in [tCMV05] to show the incompleteness of a certain hybrid logic involving the Church-Rosser formula $\Diamond\Box p \to \Box\Diamond p$.

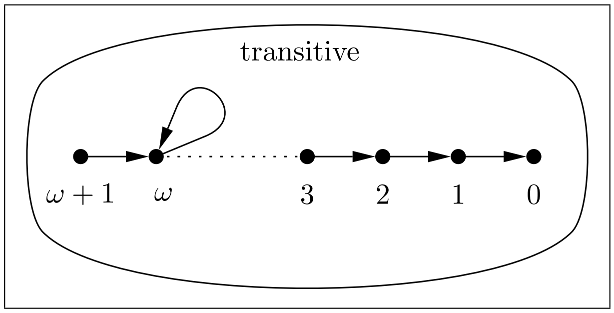

Figure 5.1: A strongly descriptive frame

We will write $\mathsf{Nom}(\mathbb{W})$ for the set of all singleton sets in $\mathbb{W}$, the notation being suggestive of the fact that, in $\mathfrak{F}$, valuations for nominals have to come from $\mathsf{Nom}(\mathbb{W})$. The elements of $\bigcup \mathsf{Nom}(\mathbb{W})$ will be referred to as the *admissible points* of $\mathfrak{F}$.

A general frame is a *reversive strongly descriptive frame* if it is reversive and strongly descriptive and satisfies clause (3) above, also with respect to the inverse relation, R^{-1}.

A formula is (locally) *sd-persistent* if it is (locally) persistent with respect to the class of all strongly descriptive frames.

Example 5.2.2 Here is an example of a strongly descriptive frame, adapted from example 2.5.2. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be the general frame with underlying Kripke frame pictured in figure 5.1. Note that ω is reflexive while all other points are irreflexive, and that the accessibility relation is transitive. Let, as in example 2.5.2, $\mathbb{W} = \{X_1 \cup X_2 \cup X_3 \mid X_i \in \mathbb{X}_i, i = 1, 2, 3\}$, where $\mathbb{X}_1$ contains all finite (possibly empty) sets of natural numbers, $\mathbb{X}_2$ contains $\emptyset$ and all sets of the form $\{x \in W \mid n \leq x \leq \omega\}$ for all $n \in \omega$, and $\mathbb{X}_3 = \{\emptyset, \{\omega + 1\}\}$. It is not difficult to check that $\mathfrak{F}$ is descriptive. Further, note that every point other than ω is admissible and that, in fact, $\mathfrak{F}$ is strongly descriptive.

So the only difference between $\mathfrak{F}$ and the general frame in example 2.5.2 is that there ω is the only successor of $\omega + 1$, the accessibility relation being not wholly transitive. That frame is descriptive but not strongly descriptive, since there $\omega + 1$, an admissible point, has no admissible successor in $\mathbb{N} \cup \{\omega\}$. $\lhd$

Ten Cate ([tC05b]) gives the following general completeness result for strongly descriptive frames.

Theorem 5.2.3 ([tC05b]) $\mathbf{K}^n \bigoplus \Sigma$*,* $\mathbf{K}^{n,u} \bigoplus \Sigma$ *and* $\mathbf{K}^{n,@} \bigoplus \Sigma$ *are strongly sound and complete with respect to the class of all strongly descriptive general frames for* Σ*, where* Σ *is any set of* $\mathcal{L}^n$*,* $\mathcal{L}^{n,u}$ *or* $\mathcal{L}^{n,@}$*-formulae, respectively.*

Corollary 5.2.4 $\mathbf{K}^n \bigoplus \Sigma$ *is strongly sound and complete with respect to the class of all Kripke frames for* Σ*, where* Σ *is any set of sd-persistent* $\mathcal{L}^n$*-formulae.*

An obvious question now arises, namely, which $\mathcal{L}^n$-formulae are sd-persistent? As it happens, not all pure formulae are sd-persistent. For example, the irreflexivity axiom, $\mathbf{j} \to \neg\Diamond\mathbf{j}$, is not locally or globally sd-persistent. Indeed, if $\mathfrak{F}$ is the strongly descriptive frame in example 5.2.2, we have $\mathfrak{F} \Vdash \mathbf{j} \to \neg\Diamond\mathbf{j}$ but $\mathfrak{F}_\sharp \not\Vdash \mathbf{j} \to \neg\Diamond\mathbf{j}$.

Lemma 5.2.5 *Let $\varphi(q_1, \ldots, q_m, \mathbf{j}, \mathbf{i}_1, \ldots, \mathbf{i}_n) \in \mathcal{L}^n_r$ be a syntactically closed formula, with $\mathsf{PROP}(\varphi) = \{q_1, \ldots, q_m\}$ and $\mathsf{NOM}(\varphi) = \{\mathbf{j}, \mathbf{i}_1, \ldots \mathbf{i}_n\}$. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame, and let $w \in W$, $Q_1, \ldots Q_m \subseteq W$ and $\{x_1\}, \ldots, \{x_m\} \subseteq W$. For the sake of brevity we will omit the parameters $Q_1, \ldots Q_m, \{x_1\}, \ldots, \{x_m\}$, and simply write $\varphi(\{v\})$ for $\varphi(Q_1, \ldots, Q_m, \{v\}, \{x_1\}, \ldots, \{x_m\})$. Then*

1. *$w \in \varphi(\{v\})$ for all $\{v\} \in \mathsf{Nom}(\mathbb{W})$ iff $w \in \varphi(\{v\})$ for all $v \in W$;*

2. *for any fixed $\{u\} \in \mathsf{Nom}(\mathbb{W})$, it holds that $w \in \varphi(\{v\})$ for all $v \in \{x \in W \mid Rux, \{x\} \in \mathsf{Nom}(\mathbb{W})\}$ iff $w \in \varphi(\{v\})$ for all $v \in \{x \in W \mid Rux\}$;*

3. *for any fixed $\{u\} \in \mathsf{Nom}(\mathbb{W})$, it holds that $w \in \varphi(\{v\})$ for all $\{v\} \in \mathsf{Nom}(\mathbb{W})$, $v \neq u$ iff $w \in \varphi(\{v\})$ for all $v \in W$, $v \neq u$.*

Proof. We proceed case by case.

1. Suppose that for all $\{v\} \in \mathsf{Nom}(\mathbb{W})$, $w \in \varphi(\{v\})$. Now for any admissible set $B \in \mathbb{W}$, it is the case that $w \in \varphi(B)$. For, since $\mathfrak{F}$ is strongly descriptive, there must be some singleton $\{b\} \in \mathsf{Nom}(\mathbb{W})$ such that $\{b\} \subseteq B$. Hence, since φ is positive in $\mathbf{j}$, we will have $w \in \varphi(B)$. Now let $x \in W$ arbitrarily. We will show that $w \in \varphi(\{x\})$. Since $\mathfrak{F}$ is descriptive all singletons are closed in $T(\mathfrak{F})$, i.e. $\{x\} = \bigcap\{B \in \mathbb{W} \mid x \in B\}$. Hence $\varphi(\{x\}) = \varphi(\bigcap\{B \in \mathbb{W} \mid x \in B\}) = \bigcap(\{\varphi(B) \mid x \in B, B \in \mathbb{W}\})$, where the last equality holds by lemma 2.5.19. The claim follows.

2. Let $\{u\} \in \mathsf{Nom}(\mathbb{W})$, and suppose that, for all $v \in \{v \in W \mid Ruv, \{v\} \in \mathsf{Nom}(\mathbb{W})\}$ it is the case that $w \in \varphi(\{v\})$. Let x be an arbitrary successor of u. We have to show that $w \in \varphi(\{x\})$. Consider any admissible set $B \in \mathbb{W}$ such that $x \in B$. By the strong descriptiveness of $\mathfrak{F}$, there must be a $\{v\} \in \mathsf{Nom}(\mathbb{W})$ such that Ruv and $v \in B$. But then, by assumption, $w \in \varphi(\{v\})$, and hence, since φ is positive in $\mathbf{j}$, we have that $w \in \varphi(B)$. The proof of this case can now be completed as in case 1, above.

3. Let $\{u\} \in \mathsf{Nom}(\mathbb{W})$, and suppose that $w \in \varphi(\{v\})$, for all $\{v\} \in \mathsf{Nom}(\mathbb{W})$, $v \neq u$. We have to show that $w \in \varphi(\{x\})$ for all $x \in W$, $x \neq u$. Let $x \in W$, $x \neq u$, arbitrarily. As before, consider any admissible set $B \in \mathbb{W}$ such that $x \in B$. But then $B - \{u\} \in \mathbb{W}$, and hence, by the strong descriptiveness of $\mathfrak{F}$ there must be a $\{b\} \in \mathbb{W}$ such that $b \in B - \{u\}$. Hence, since φ is positive in $\mathbf{j}$, we have that $w \in \varphi(B)$. Again the proof can now be completed as in case 1, above.

QED

Proposition 5.2.6 *Any positive, syntactically closed $\mathcal{L}^n_r$-formula is locally sd-persistent.*

Proof. Let $\varphi(p_1, \ldots, p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n) \in \mathcal{L}^n_r$ be syntactically closed and positive in all propositional variables, with $\mathsf{PROP}(\varphi) = \{p_1, \ldots, p_m\}$ and $\mathsf{NOM}(\varphi) = \{\mathbf{i}_1, \ldots \mathbf{i}_n\}$. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame for $\mathcal{L}^n_\tau$, and $w \in W$ a point in $\mathfrak{F}$. Note that, by the upward monotonicity of φ in $p_1, \ldots, p_m$,

$$(\mathfrak{F}, w) \Vdash \varphi(p_1, \ldots, p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n) \text{ iff } (\mathfrak{F}, w) \Vdash \varphi(\bot/p_1, \ldots, \bot/p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n).$$

So assume that $\mathfrak{F} \Vdash \varphi(\bot/p_1, \ldots, \bot/p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n)$. We have to show that

$$(\mathfrak{F}_\sharp, w) \Vdash \varphi(\bot/p_1, \ldots, \bot/p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n),$$

where $\mathfrak{F}_\sharp = (W, R)$ is the underlying Kripke frame of $\mathfrak{F}$. We will henceforth omit the parameters '$\bot/p_i$' and merely write $\varphi(\mathbf{i}_1, \ldots, \mathbf{i}_n)$.

By assumption, it is the case that $w \in \varphi(\{a_1\}, \ldots, \{a_n\})$ for all admissible singletons $\{a_1\}, \ldots \{a_n\} \in \mathbb{W}$. By applying lemma 5.2.5 to the first coordinate, we find that $w \in \varphi(\{x_1\}, \ldots, \{a_n\})$ for *all* $x_1 \in W$ and all $\{a_2\}, \ldots \{a_n\} \in \mathbb{W}$. Now, by proceeding inductively and applying lemma 5.2.5 to each coordinate, we see that $w \in \varphi(\{x_1\}, \ldots, \{x_n\})$ for *all* $x_1, \ldots, x_n \in W$. It follows that

$$(\mathfrak{F}_\sharp, w) \Vdash \varphi(\bot/p_1, \ldots, \bot/p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n),$$

and hence that

$$(\mathfrak{F}_\sharp, w) \Vdash \varphi(p_1, \ldots, p_m, \mathbf{i}_1, \ldots, \mathbf{i}_n).$$

QED

Corollary 5.2.7 *Any pure, syntactically closed formula from $\mathcal{L}^n_r$ is locally sd-persistent.*

Proposition 5.2.6 and corollary 5.2.7 are too restrictive for our purposes — we have to be able to guarantee sd-persistence of formulae with ceratin negative nominal occurrences as well. To that aim we make the some definitions and prove some lemmas:

Definition 5.2.8 A formula of the form $\mathbf{j} \to \Diamond\mathbf{k}$ or $\neg\mathbf{j} \vee \Diamond\mathbf{k}$ is called *diamond-link formula*, i.e. diamond-link formulae are the translations of diamond-link sequents. A *general diamond-link formula* is any conjunction of diamond-link formulae.

Given a general diamond-link formula φ, the *dependency digraph of* φ is the directed graph $\langle V_\varphi, E_\varphi \rangle$, with vertex set V_φ consisting of all nominals occurring in φ, and edge set E_φ, such that $(\mathbf{j}, \mathbf{k}) \in E_\varphi$ iff $\mathbf{j} \to \Diamond\mathbf{k}$ (or $\neg\mathbf{j} \vee \Diamond\mathbf{k}$) is a conjunct of φ. The dependency digraph of a general diamond-link formula is *tree-like*, if it is a tree, in other words, if it has a root, i.e. a vertex from which every other vertex is accessible along a directed path, and every vertex other than the root is reachable from the root along exactly one directed path. A dependency digraph is called *forest-like* if its is the disjoint union of tree-like dependency digraphs. A general diamond link-formula is *tree-like* (*forest-like*) if its dependency digraph is tree-like (forest-like).

Lemma 5.2.9 *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive general frame, and φ a forest-like general diamond-link formula. Suppose $\mathsf{NOM}(\varphi) = \{\mathbf{j}_1, \ldots, \mathbf{j}_n\}$. Then for all $x_1, \ldots, x_n \in W$ and $X_1, \ldots, X_n \in \mathbb{W}$ with $x_i \in X_i$, $1 \leq i \leq n$ such that*

$$\mathfrak{F}_\sharp \Vdash \varphi[\mathbf{j}_1 := x_1, \ldots, \mathbf{j}_n := x_n],$$

there exist admissible singletons $\{a_1\}, \ldots, \{a_n\} \in \mathsf{Nom}(\mathbb{W})$ such that $a_i \in X_i$, $1 \leq i \leq n$ and

$$\mathfrak{F} \Vdash \varphi[\mathbf{j}_1 := a_1, \ldots, \mathbf{j}_n := a_n].$$

Proof. We illustrate the proof-idea with an example. The general proof, which is made extremely tedious by the technical overhead of keeping track of the tree structures, should then be clear. Consider the formula

$$\begin{aligned} & [(\mathbf{j}_1 \to \Diamond \mathbf{j}_2) \wedge (\mathbf{j}_1 \to \Diamond \mathbf{j}_3) \wedge (\mathbf{j}_2 \to \Diamond \mathbf{j}_4) \wedge (\mathbf{j}_3 \to \Diamond \mathbf{j}_5) \wedge (\mathbf{j}_3 \to \Diamond \mathbf{j}_6) \wedge (\mathbf{j}_3 \to \Diamond \mathbf{j}_7)] \\ \wedge \quad & [(\mathbf{k}_1 \to \Diamond \mathbf{k}_2) \wedge (\mathbf{k}_1 \to \Diamond \mathbf{k}_3) \wedge (\mathbf{k}_3 \to \Diamond \mathbf{k}_4)]. \end{aligned}$$

The dependency digraph of this formula is illustrated in figure 5.2. Suppose

$$\mathfrak{F}_\sharp \Vdash \varphi[\mathbf{j}_1 := x_1, \ldots, \mathbf{j}_7 := x_7, \mathbf{k}_1 := y_1, \ldots, \mathbf{k}_4 := y_4],$$

hence Rx_1x_2, Rx_1x_3, Rx_2x_4, Rx_3x_5, Rx_3x_6, and Rx_3x_7, and Ry_1y_2, Ry_1y_3, and Ry_3x_4. Let $X_1, \ldots, X_7, Y_1, \ldots, Y_7 \in \mathbb{W}$ such that $x_i \in X_i$, $1 \leq i \leq 7$ and $y_i \in Y_i$, $1 \leq i \leq 4$. But then

$$X_1 \cap \Diamond(X_2 \cap \Diamond(X_4)) \cap \Diamond(X_3 \cap \Diamond(X_5) \cap \Diamond(X_6) \cap \Diamond(X_7)) \neq \emptyset,$$

and

$$Y_1 \cap \Diamond(Y_2) \cap \Diamond(Y_3 \cap \Diamond Y_4) \neq \emptyset.$$

Appealing to the strong descriptiveness of $\mathfrak{F}$, there exists an admissible $\{a_1\} \subseteq X_1 \cap \Diamond(X_2 \cap \Diamond(X_4)) \cap \Diamond(X_3 \cap \Diamond(X_5) \cap \Diamond(X_6) \cap \Diamond(X_7))$, and hence admissible $\{a_2\} \subseteq X_2 \cap \Diamond(X_4)$ and $\{a_3\} \subseteq X_3 \cap \Diamond(X_5) \cap \Diamond(X_6) \cap \Diamond(X_7)$ such that Ra_1a_2 and Ra_1a_3. Again by the strong descriptiveness of $\mathfrak{F}$, there are admissible $\{a_i\} \subseteq X_i$, $i = 4, 5, 6, 7$, such that Ra_2a_4, Ra_3a_5, Ra_3a_6 and Ra_3a_7. Similarly we find admissible singletons $\{b_i\} \subseteq Y_i$, $1 \leq i \leq 4$, such that Rb_1b_2, Rb_1b_3, and Rb_3b_4. Hence

$$\mathfrak{F} \Vdash \varphi[\mathbf{j}_1 := a_1, \ldots, \mathbf{j}_7 := a_7, \mathbf{k}_1 := b_1, \ldots, \mathbf{k}_4 := b_4].$$

QED

The next proposition will be used in proving the sd-persistence of formulae on which the modified version of SQEMA, introduced in the next subsection, succeeds.

Proposition 5.2.10 *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame. Any formula of the form $\varphi \wedge \psi \in \mathcal{L}_r^n$ with ψ pure and syntactically open, and φ a forest-like general diamond-link formula, is globally satisfiable on $\mathfrak{F}$ iff it is globally satisfiable on $\mathfrak{F}_\sharp$.*

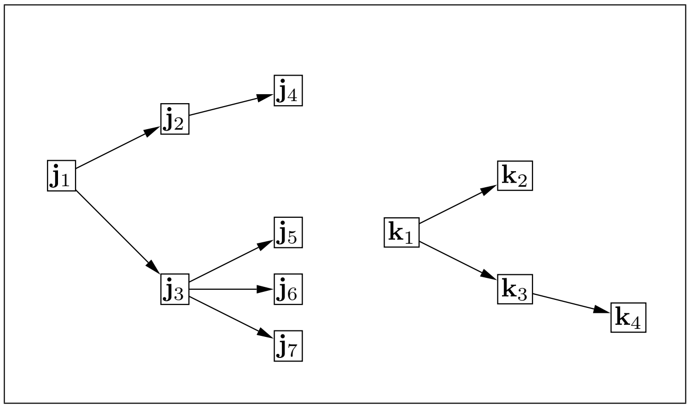

Figure 5.2: The dependency digraph of the general diamond-link formula in lemma 5.2.9

Proof. Suppose that $\mathsf{NOM}(\varphi) \subseteq \{\mathbf{j}_1, \ldots, \mathbf{j}_n\}$ and that $\mathsf{NOM}(\psi) - \mathsf{NOM}(\varphi) \subseteq \{\mathbf{k}_1, \ldots, \mathbf{k}_m\}$. The implication from left to right is trivial. For the sake of the other direction, suppose that there are $x_1, \ldots x_n, y_1, \ldots, y_m \in W$ such that

$$\varphi(\{x_1\}, \ldots, \{x_n\}) = W \quad \text{and} \quad \psi(\{x_1\}, \ldots, \{x_n\}, \{y_1\}, \ldots, \{y_m\}) = W.$$

Hence $\neg\psi(\{x_1\}, \ldots, \{x_n\}, \{y_1\}, \ldots, \{y_m\}) = \emptyset$. But then, since singletons are closed in descriptive frames,

$$\neg\psi(\bigcap \mathbb{X}_1, \ldots, \bigcap \mathbb{X}_n, \bigcap \mathbb{Y}_1, \ldots, \bigcap \mathbb{Y}_m) = \emptyset,$$

where, $\mathbb{X}_i = \{X \in \mathbb{W} \mid x_i \in X\}$, $1 \leq i \leq n$, and $\mathbb{Y}_i = \{Y \in \mathbb{W} \mid y_i \in Y\}$, $1 \leq i \leq m$. Hence, by lemma 2.5.19 and the fact that $\neg\psi$ is syntactically closed, we have

$$\bigcap_{X_1 \in \mathbb{X}_1} \cdots \bigcap_{X_n \in \mathbb{X}_n} \bigcap_{Y_1 \in \mathbb{Y}_1} \cdots \bigcap_{Y_m \in \mathbb{Y}_m} \neg\psi(X_1, \ldots, X_n, Y_1, \ldots, Y_m) = \emptyset.$$

By compactness there exists $X_1, \ldots, X_n \in \mathbb{W}$ and $Y_1, \ldots, Y_m \in \mathbb{W}$ with $x_i \in X_i$, $1 \leq i \leq n$, and $y_i \in Y_i$, $1 \leq i \leq m$, such that

$$\neg\psi(X_1, \ldots, X_n, Y_1, \ldots, Y_m) = \emptyset.$$

By lemma 5.2.9 there are $a_1, \ldots, a_n, \in \bigcup \mathsf{Nom}(\mathbb{W})$ such that $a_i \in X_i$, $1 \leq i \leq n$ and

$$\varphi(\{a_1\}, \ldots, \{a_n\}) = W.$$

Since $\neg\psi$ is positive in all nominals, we can choose arbitrary $b_1, \ldots, b_n \in \bigcup \mathsf{Nom}(\mathbb{W})$ such that $b_i \in Y_i$, $1 \leq i \leq m$ and $\neg\psi(\{a_1\}, \ldots, \{a_n\}, \{b_1\}, \ldots, \{b_m\}) = \emptyset$. Hence

$$\varphi(\{a_1\}, \ldots, \{a_n\}) \cap \psi(\{a_1\}, \ldots, \{a_n\}, \{b_1\}, \ldots, \{b_m\}) = W.$$

In other words, $\varphi \wedge \psi$ is globally satisfiable on $\mathfrak{F}$. QED

The following corollary, for parameterized satisfiability, is easy to prove:

Corollary 5.2.11 *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame. Any formula of the form $\varphi \wedge \psi \in \mathcal{L}_r^n$ with ψ pure and syntactically open, and φ a forest-like general diamond-link formula, is globally satisfiable with parameters from $\bigcup \mathsf{Nom}(\mathbb{W})$ on $\mathfrak{F}$ iff it is globally satisfiable on $\mathfrak{F}_\sharp$ with the same parameters.*

5.2.2 Adapting SQEMA to prove sd-persistence

In this section we develop a variant of SQEMA, called SQEMA^{sd}, which is specifically adapted to work on strongly descriptive frames. Our aim is to adapt SQEMA in such a way that it would transform input formulae into pure formulae of the form described in proposition 5.2.10. This will enable us to prove the sd-persistence of the $\mathcal{L}_r^n$-formulae on which the adapted version succeeds. Since proposition 5.2.10 does not allow for arbitrary nominal occurrences, the adapted algorithm will have to be able to eliminate not only propositional variables, but also 'bad' nominal occurrences. We begin by providing an Ackermann-type lemma — essentially a triviality — that will facilitate the elimination of such nominal occurrences.

Lemma 5.2.12 (Ackermann's Lemma for Nominals) *Let $B \in \mathcal{L}_r^n$, and let $\mathbf{j}$ and $\mathbf{k}$ be nominals. Then for any model $\mathcal{M} = (\mathfrak{F}, V)$ based on a strongly descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$, it holds that*

$$(\mathfrak{F}, V) \Vdash B(\mathbf{j}/\mathbf{k})$$

if and only if there exists an admissible valuation V', differing from V at most in the valuation of $\mathbf{k}$, such that

$$(\mathfrak{F}, V') \Vdash (\mathbf{j} \to \mathbf{k}) \wedge B(\mathbf{k}).$$

The following transformation rule can now be formulated:

Ackermann-Rule for Nominals:

$$\text{The system} \quad \left\|\begin{array}{l} \mathbf{j} \Rightarrow \mathbf{k}, \\ B_1(\mathbf{k}), \\ \vdots \\ B_m(\mathbf{k}), \end{array}\right. \quad \text{is replaced by} \quad \left\|\begin{array}{l} B_1[\mathbf{j}/\mathbf{k}], \\ \vdots \\ B_m[\mathbf{j}/\mathbf{k}]. \end{array}\right.$$

where each of $B_1, \ldots, B_m$ is negative in $\mathbf{k}$.

We note that, for the rule to be applicable there can be only one positive occurrence of $\mathbf{k}$ in the system, and that the soundness of the rule on strongly descriptive frames follows from lemma 5.2.12. Of course, lemma 5.2.12 would guarantee the soundness of a stronger rule, viz. a rule that does not require the B_i to be negative in $\mathbf{k}$. Such a rule would be too general for our purposes, however.

Given an execution of SQEMA on an $\mathcal{L}^n$ input formula φ, we will refer to the elements of $\mathsf{NOM}(\varphi)$ as the *input nominals* of the execution, and to all other nominals appearing in the systems during the execution (i.e. the reserved nominal $\mathbf{i}$ as well as all nominals introduced by the $\diamond$-rule) as the *introduced nominals* of the execution.

The algorithm SQEMA^{sd} is obtained from the basic algorithm SQEMA, by

1. adding the Ackermann-rule for nominals to the set of transformation rules, and
2. requiring that, apart from the elimination of all occurring propositional variables, all input-nominals with positive occurrences in the system are to be eliminated as well.

Definition 5.2.13 An $\mathcal{L}^n_r$-formula is *inverse existential* if all occurrences of $\diamond^{-1}$ in it are positive and all occurrences of $\Box^{-1}$ are negative. Equivalently, when rewritten in negation normal form, there are no occurrences of $\Box^{-1}$ in the formula. A $\mathcal{L}^n_r$-formula is *inverse universal* if all occurrences of $\Box^{-1}$ in it are positive and all occurrences of $\diamond^{-1}$ are negative. Equivalently, when rewritten in negation normal form, there are no occurrences of $\diamond^{-1}$ in the formula. Clearly, negation maps inverse existential formulae to inverse universal formulae and vice versa.

Observe that, if $\varphi(p_1, \ldots, p_n, \mathbf{j}_1, \ldots, \mathbf{j}_m) \in \mathcal{L}^n_r$ is inverse existential then, for any fresh propositional variables $q_1, \ldots, q_m$, the formula $\varphi(p_1, \ldots, p_n, q_1/\mathbf{j}_1, \ldots, q_m/\mathbf{j}_m)$ is syntactically closed. Hence, given any strongly descriptive frame $\mathfrak{F} = (W, R, \mathbb{W})$ and sets $X_1, \ldots, X_n \in \mathbb{W}$, $\{x_1\}, \ldots, \{x_m\} \in \mathsf{Nom}(\mathbb{W})$, we have, by lemma 2.5.17, that $\varphi(X_1, \ldots, X_n, \{x_1\}, \ldots, \{x_m\})$ will be a closed set with respect to $T(\mathfrak{F})$. Taking this observation further, it is not difficult to see that the following analogues of lemmas 2.5.19 and 2.5.20 hold.

Lemma 5.2.14 (Esakia's lemma for inverse existential formulae on sd-frames) *Let $\varphi(\overline{q}, p, \overline{\mathbf{j}}) \in \mathcal{L}^n_r$ be an inverse existential formula, positive in p, $\mathfrak{F} = (W, R, \mathbb{W})$ a strongly descriptive frame, and $\{C_i\}_{i \in I}$ a downwards directed family of closed sets from $\mathbf{C}(\mathfrak{F})$. Then, for any $\overline{Q} \in \mathbb{W}$ and $\overline{a} \in \mathsf{Nom}(\mathbb{W})$,*

$$\varphi(\overline{Q}, \bigcap_{i \in I} C_i, \overline{a}) = \bigcap_{i \in I} \varphi(\overline{Q}, C_i, \overline{a}).$$

Lemma 5.2.15 (Restricted version of Ackermann's lemma for sd-frames) *Let $\mathfrak{F} = \langle W, R, \mathbb{W} \rangle$ be a strongly descriptive frame, and $A(\overline{q}, \overline{\mathbf{j}})$ and $B(\overline{q}, p, \overline{\mathbf{j}})$ inverse existential and universal $\mathcal{L}^n_r$-formulae, respectively, with p not occurring in A and B negative in p. Then, for all $\overline{Q} \in \mathbb{W}$, $\overline{a} \in \mathsf{Nom}(\mathbb{W})$, it holds that*

$$B(\overline{Q}, A(\overline{Q}, \overline{a}), \overline{a}) = W$$

if and only if there is a $P \in \mathbb{W}$ such that

$$A(\overline{Q}, \overline{a}) \subseteq P \text{ and } B(\overline{Q}, P, \overline{a}) = W.$$

Adapting definition 2.5.7, whilst bearing in mind that strongly descriptive frames contain sufficiently many singletons for the proper interpretation of $\mathcal{L}^n$, we obtain the following version of transformation equivalence:

Definition 5.2.16 Formulae $\varphi, \psi \in \mathcal{L}^n_r$ are *transformation equivalent on strongly descriptive frames* if, for every model $\mathcal{M} = (\mathfrak{F}, V)$ based on a strongly descriptive frame $\mathfrak{F}$, such that $\mathcal{M} \Vdash \varphi$ there exists a $(\mathsf{PROP}(\varphi) \cap \mathsf{PROP}(\psi), \mathsf{NOM}(\varphi) \cap \mathsf{NOM}(\psi))$-related model $\mathcal{M} = (\mathfrak{F}, V')$ based on $\mathfrak{F}$, such that $\mathcal{M}' \Vdash \psi$, and vice versa. We will write $\varphi \equiv^{sd}_{\text{trans}} \varphi$ if φ and ψ are transformation equivalent on strongly descriptive frames.

Lemma 5.2.17 (Soundness for Strongly Descriptive Frames) *Let Sys' be a system obtained from a system Sys by the application of a transformation rule of SQEMA^{sd}. Then $\mathsf{Form}(\mathsf{Sys}) \equiv^{sd}_{\text{trans}} \mathsf{Form}(\mathsf{Sys}')$.*

Proof. We have to show that the transformation rules of SQEMA^{sd} preserve transformation equivalence on strongly descriptive frames. This is easily justified. In particular, the case for the (ordinary) Ackermann-rule follows from lemma 5.2.15 with the help of claim 1, below.

Claim 1 During the entire (successful or unsuccessful) execution of SQEMA^{sd} on an $\mathcal{L}^n$ input formula, the antecedents of all sequents are inversely existential $\mathcal{L}^n_r$-formulae, while the consequents of all sequents are inversely universal $\mathcal{L}^n_r$-formulae.

Proof of Claim This follows from the proof of lemma 2.5.21. ◀

The Ackermann-rule for nominals is justified by lemma 5.2.12. We also check the $\Diamond$-rule. Indeed, suppose that $\mathsf{Form}(\mathsf{Sys})$ is of the form $(\neg\mathbf{j} \vee \Diamond\varphi) \wedge \psi$. Let $\mathfrak{F} = (W, R, \mathbb{W})$ be any strongly descriptive frame. Then $(\mathfrak{F}, V) \Vdash ((\neg\mathbf{j} \vee \Diamond\varphi) \wedge \psi)[\mathbf{i} := w]$ iff $[\![\psi]\!]_{(\mathfrak{F},V)} = W$ and $\{a\} = [\![\mathbf{j}]\!]_{(\mathfrak{F},V)} \subseteq [\![\Diamond\varphi]\!]_{(\mathfrak{F},V)}$, where $V(\mathbf{j}) = \{a\} \in \mathsf{Nom}(\mathbb{W})$. By claim 1, $\Diamond\varphi$ (and hence φ) is inversely universal. Hence $[\![\varphi]\!]_{(\mathfrak{F},V)}$ is an open set with respect to $T(\mathfrak{F})$, i.e. $[\![\varphi]\!]_{(\mathfrak{F},V)} = \bigcup\{C \in \mathbb{W} \mid [\![\varphi]\!]_{(\mathfrak{F},V)} \subseteq C\}$. Hence, by the strong descriptiveness of $\mathfrak{F}$, there exists a $\{b\} \in \mathsf{Nom}(\mathbb{W})$ such that $b \in [\![\varphi]\!]_{(\mathfrak{F},V)}$ and such that Rab. It follows that $(\mathfrak{F}, V) \Vdash ((\neg\mathbf{j} \vee \Diamond\varphi) \wedge \psi)[\mathbf{i} := w]$ iff the valuation of V may be changed, at most in its assignment to the fresh nominal $\mathbf{k}$, to obtain V' such that $V'(\mathbf{k}) = b$ and hence $(\mathfrak{F}, V') \Vdash ((\neg\mathbf{j} \vee \Diamond\mathbf{k}) \wedge (\neg\mathbf{k} \vee \varphi) \wedge \psi)[\mathbf{i} := w]$. QED

Example 5.2.18 Consider the execution of SQEMA^{sd} on the formula $\Diamond(\mathbf{j} \wedge \Box p) \to (\Diamond\mathbf{k} \vee \Box(\mathbf{j} \vee \Diamond p))$. After negating and applying the $\wedge$-rule, the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond(\mathbf{j} \wedge \Box p) \\ \mathbf{i} \Rightarrow (\Box\neg\mathbf{k} \wedge \Diamond(\neg\mathbf{j} \wedge \Box\neg p)) \end{array} \right.$$

is obtained. Applying the $\Diamond$ and $\wedge$-rules followed by the $\Box$-rule and then the Ackermann-rule (with respect to p) produces

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{l} \\ \mathbf{l} \Rightarrow \mathbf{j} \\ \mathbf{i} \Rightarrow (\Box\neg\mathbf{k} \wedge \Diamond(\neg\mathbf{j} \wedge \Box\Box^{-1}\neg\mathbf{l})) \end{array} \right. .$$

Although the system obtained is now pure, it still contains a positive occurrence of the input nominal $\mathbf{j}$. The Ackermann-rule for nominals is now applied to obtain the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{l} \\ \mathbf{i} \Rightarrow (\Box\neg\mathbf{k} \wedge \Diamond(\neg\mathbf{l} \wedge \Box\Box^{-1}\neg\mathbf{l})) \end{array} \right. .$$

After negation we obtain the formula $(\mathbf{i} \wedge \Box\neg\mathbf{l}) \vee (\mathbf{i} \wedge (\Diamond\mathbf{k} \vee \Box(\mathbf{l} \vee \Diamond\Diamond^{-1}\mathbf{l})))$, which, we note, is of the form prescribed by proposition 5.2.10. This formula can be rewritten as $\mathbf{i} \wedge (\Diamond\mathbf{k} \vee [\Box\neg\mathbf{l} \vee \Box(\mathbf{l} \vee \Diamond\Diamond^{-1}\mathbf{l})]$. After translation and simplification this becomes

$$(\forall y Rxy) \vee (\forall y \forall z(Rxy \wedge Rxz \to (y \neq z \to \exists u(Ryu \wedge Rzu)))).$$

Hence a point in a Kripke (or strongly descriptive) frame validates the input formula $\Diamond(\mathbf{j} \wedge \Box p) \to (\Diamond\mathbf{k} \vee \Box(\mathbf{j} \vee \Diamond p))$, iff it is either a spy point (i.e. a point from which all points in the frame are accessible) or satisfies (locally) a weakened version of the Church-Rosser property. This property is definable (on Kripke frames) neither by an $\mathcal{L}$-formula nor by a pure $\mathcal{L}^n$-formula. Indeed, the property is undefinable by a modal formula since it is not invariant under disjoint unions. The undefinability by pure formulae may be seen by considering the frames used in [GG93] (see also [tC05b]) to show that the Church-Rosser property is undefinability by pure formulae, even with the help of the universal modality. $\lhd$

The next useful proposition shows that a formula $\varphi \in \mathcal{L}^n$ is valid in a strongly descriptive frame whenever it is valid at every admissible point in that frame.

Proposition 5.2.19 *Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame and $\varphi \in \mathcal{L}^n_r$ an inversely existential formula. Then $\mathfrak{F} \Vdash \varphi$ whenever $(\mathfrak{F}, w) \Vdash \varphi$ for all $w \in \bigcup \mathsf{Nom}(\mathbb{W})$.*

Proof. Suppose that $(\mathfrak{F}, w) \Vdash \varphi(\overline{p}, \overline{\mathbf{k}})$ for all $w \in \bigcup \mathsf{Nom}(\mathbb{W})$. Suppose further, by way of contradiction, that for some $v \in W$, some $\overline{P} \in \mathbb{W}$ and some $\overline{a} \in \mathsf{Nom}(\mathbb{W})$, we have $v \notin \varphi(\overline{P}, \overline{a})$, i.e. $\{v\} \cap \varphi(\overline{P}, \overline{a}) = \emptyset$. But, since singletons are closed in descriptive frames, $\{v\} = \bigcap\{C \in \mathbb{W} \mid v \in C\}$. Moreover, as remarked above, since φ is inversely existential, $\varphi(\overline{P}, \overline{a})$ is also closed with respect to $T(\mathfrak{F})$, so $\varphi(\overline{P}, \overline{a}) = \bigcap\{D \in \mathbb{W} \mid \varphi(\overline{P}, \overline{a}) \subseteq D\}$. So we have $\bigcap\{C \in \mathbb{W} \mid v \in C\} \cap \bigcap\{D \in \mathbb{W} \mid \varphi(\overline{P}, \overline{a}) \subseteq D\} = \emptyset$. Hence, by compactness, there exist admissible sets $C, D \in \mathbb{W}$ with $v \in C$ and $\varphi(\overline{P}, \overline{a}) \subseteq D$, such that $C \cap D = \emptyset$. But then $C \cap \varphi(\overline{P}, \overline{a}) = \emptyset$. But there is at least once admissible point, say c, such that $c \in C$, and hence $c \notin \varphi(\overline{P}, \overline{a})$, which contradicts our assumption that $(\mathfrak{F}, w) \Vdash \varphi$ for all $w \in \bigcup \mathsf{Nom}(\mathbb{W})$. QED

Theorem 5.2.20 *Every formula $\varphi \in \mathcal{L}^n$ on which SQEMA^{sd} succeeds is globally sd-persistent.*

Proof. Suppose SQEMA^{sd} succeeds on $\varphi \in \mathcal{L}^n$. We use the following claim:

CLAIM 1 The formula $\mathsf{pure}(\varphi)$ (obtained in step Postprocessing.2 of the execution) is of the form $\bigvee_{i=1}^n \psi_i$ for some $n \in \mathbb{N}^+$ and with the each ψ_i either

1. a syntactically open formula,
2. a forest-like general diamond-link formula, or
3. a conjunction of a syntactically open formula and a forest-like general diamond-link formula.

Proof of Claim The ψ_i are the formulae $\mathsf{Form}(\mathsf{Sys}_i)$ for the final systems Sys_i on the n disjunctive branches of the execution. It is easy to verify that in each $\mathsf{Form}(\mathsf{Sys}_i)$, (i) each introduced nominal has exactly one positive occurrence, namely in the translation of the diamond-link formula introduced at its introduction (by induction on the application of transformation rules); (ii) no input nominal occurs positively in pure (by the assumption of success); (iii) $\mathsf{Form}(\mathsf{Sys}_i)$ is inversely universal (see the proof of lemma 5.2.17). We deduce that each ψ_i is indeed either a syntactically open formula, a general diamond-link formula, or a conjunction of such formulae.

It only remains to verify that in each ψ_i that contains a general diamond-link formula as a conjunct, that general diamond-link formula is forest-like. But this follows once we note that every nominal occurring positively in a diamond link-formula is introduced by the $\Diamond$-rule, and that this rule only introduces new nominals, not occurring in the system yet. ◀

Let $\mathfrak{F} = (W, R, \mathbb{W})$ be a strongly descriptive frame. Proceeding exactly as in the proof of 2.5.23 and using lemma 5.2.17, we see that, for all $w \in \bigcup \mathsf{Nom}(\mathbb{W})$ it is the case that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$. Similarly, for all $w \in W$ it holds that $(\mathfrak{F}_\sharp, w) \Vdash \varphi$ iff $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}_\sharp$.(In the case of the general frame $\mathfrak{F}$, we have to restrict to $w \in \bigcup \mathsf{Nom}(\mathbb{W})$, for we cannot assign non-admissible singletons to $\mathbf{i}$.)

Hence $\mathfrak{F} \Vdash \varphi$ iff (by lemma 5.2.19) $(\mathfrak{F}, w) \Vdash \varphi$ for all $w \in \bigcup \mathsf{Nom}(\mathbb{W})$, iff $\mathsf{pure}(\varphi)$ is not globally satisfiable on $\mathfrak{F}$ iff (by lemma 5.2.10 and claim 1) $\mathsf{pure}(\varphi)$ is not globally satisfiable on $\mathfrak{F}_\sharp$ iff $\mathsf{pure}(\varphi)$ is not globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}_\sharp$ for any $w \in W$ iff $\mathfrak{F}_\sharp \Vdash \varphi$. QED

Corollary 5.2.21 $\mathbf{K}^n \oplus \varphi$ *is strongly sound and complete (with respect to its class of Kripke frames) whenever* SQEMA^{sd} *succeeds on* $\varphi \in \mathcal{L}^n$.

5.2.3 Syntactic classes

In the previous subsection we have seen that, apart from computing first-order local frame equivalents, SQEMA^{sd} also guarantees the sd-persistence of the $\mathcal{L}^n$-formulae on which it succeeds, and hence the completeness of the extensions of $\mathbf{K}^n$ axiomatized by these formulae. In this subsection we evaluate SQEMA^{sd}'s performance on syntactically specified classes of elementary and/or persistent $\mathcal{L}^n$-formulae.

The basic modal language

First let us consider the familiar syntactically specified classes of $\mathcal{L}$-formulae. SQEMA^{sd} is an extension of SQEMA with a new rule, plus the requirement that positive occurrences of input nominals have to be eliminated. Since there are no input nominals in the case of $\mathcal{L}$-input formulae, it follows that SQEMA^{sd} succeeds on all $\mathcal{L}$-formulae on which SQEMA succeeds. In particular, we have the following theorem.

Theorem 5.2.22 SQEMA^{sd} *succeeds on all* $\mathcal{L}$*-formulae on which* SQEMA *succeeds.*

The next proposition could be seen as a corollary of theorem 5.2.22, but it in fact follows immediately form the d-persistence of the members of these classes and the fact that all d-persistent formulae are sd-persistent.

Proposition 5.2.23 *All Sahlqvist, Sahlqvist–van Benthem and monadic inductive $\mathcal{L}$-formulae are sd-persistent.*

Combining this with theorem 5.2.3 we obtain the following theorem, subsuming the result from [tCMV05, tC05b], which holds for Sahlqvist formulae.

Theorem 5.2.24 $\mathbf{K}^n \oplus \Sigma$ *is strongly sound and complete with respect to its class of Kripke frames whenever* Σ *is a set of Sahlqvist, Sahlqvist–van Benthem and/or monadic inductive $\mathcal{L}$-formulae.*

Adding nominals

From [GPT87] it is know that all extensions of $\mathbf{K}^n$ with pure $\mathcal{L}^n$-formulae are strongly complete (theorem 0.2.2). In [tCMV05, tC05b] it is proved that all extensions of $\mathbf{K}^n$ with Sahlqvist axioms from the basic language $\mathcal{L}$ are complete. It is also shown in [tCMV05, tC05b] that these two results cannot be combined — the logic obtained by adding the axiom $\varphi = (\Diamond\Box p \to \Box\Diamond p) \wedge (\Diamond(\mathbf{i} \wedge \Diamond\mathbf{j}) \to \Box(\Diamond\mathbf{j} \to \mathbf{i}))$ to $\mathbf{K}^n$ is incomplete. As far as the author is aware, this sums up everything concerning general completeness results for syntactically specified classes of $\mathcal{L}^n$-formulae that is available in the literature.

In the rest of this section we define a new class of $\mathcal{L}^n$-formulae on which SQEMA^{sd} succeeds. It then follows that the members of this class axiomatize complete extensions of $\mathbf{K}^n$. It will be seen that this class subsumes and extends the known results.

Definition 5.2.25 An $\mathcal{L}^n$-formula φ in negation normal form is a *nominalized Sahlqvist–van Benthem formula* if it satisfies the following conditions:

(NSB1) For every occurring propositional variable p, either

- **(NSB1.1)** there is no positive occurrence of p in a subformula $\psi \wedge \chi$ or $\Box\psi$ which is in the scope of a $\Diamond$, or
- **(NSB1.2)** there is no negative occurrence of p in a subformula $\psi \wedge \chi$ or $\Box\psi$ which is in the scope of a $\Diamond$.

(NSB2) No negative nominal occurrence in φ is in the scope of a $\Diamond$.

(NSB3) Every two negative occurrence of a given nominal $\mathbf{j}$, are in a subformula of the form $\psi \wedge \chi$, such that one occurrence is in ψ and the other is in χ.

In other words, a nominalized Sahlqvist–van Benthem formula is a formula in negation normal form which satisfies all constraints on propositional variables imposed on Sahlqvist–van Benthem formulae, and in which positive nominals are allowed to occur arbitrarily, while negative nominals may not occur in the scope of diamonds and every two have to be separated by a conjunction.

Example 5.2.26 The formula $\varphi = (\Diamond\Box p \to \Box\Diamond p) \wedge (\Diamond(\mathbf{i} \wedge \Diamond\mathbf{j}) \to \Box(\Diamond\mathbf{j} \to \mathbf{i}))$ from [tCMV05, tC05b] becomes

$$(\Box\Diamond\neg p \vee \Box\Diamond p) \wedge (\Box(\neg\mathbf{i} \vee \Box\neg\mathbf{j}) \vee \Box(\Box\neg\mathbf{j} \vee \mathbf{i})),$$

after being rewritten in negation normal form. Although its satisfies conditions (NSB1) and (NSB2), this formula fails to be a nominalized Sahlqvist–van Benthem formula since the two negative occurrences of $\mathbf{j}$ are not separated by a conjunction.

The formula $\Diamond(\mathbf{j} \wedge \Box p) \to (\Diamond\mathbf{k} \vee \Box(\mathbf{j} \vee \Diamond p))$ from example 5.2.18 becomes

$$\Box(\neg\mathbf{j} \vee \Diamond\neg p) \vee (\Diamond\mathbf{k} \vee \Box(\mathbf{j} \vee \Diamond p))$$

when rewritten in negation normal form. This is a nominalized Sahlqvist–van Benthem formula. Note that it contains only one negative nominal occurrence, and that this occurrence is not in the scope of a diamond.

The irreflexivity axiom, $\mathbf{j} \to \neg\Diamond\mathbf{j}$ is not a nominalized Sahlqvist–van Benthem formula, since in $\neg\mathbf{j} \vee \Box\neg\mathbf{j}$ the two negative occurrences of $\mathbf{j}$ are not separated by a conjunction. Recall that it was shown above that this axiom is not sd-persistent. $\lhd$

We will show that SQEMA^{sd} succeeds on every nominalized Sahlqvist–van Benthem formula, and hence that every such formula is sd-persistent. To that purpose we formulate a definition which can be viewed as a (smoothed out) analogue of the definition of a simple dual Sahlqvist–van Benthem system (definition 2.6.1).

Definition 5.2.27 Call a system Sys a *simple dual nominalized Sahlqvist–van Benthem system*, or an SDNS for short, if each sequent in Sys has the from $\mathbf{j} \Rightarrow \psi$, where $\mathbf{j}$ is a nominal, and the following conditions are satisfied:

(SDNS1) In $\mathsf{Form}(\mathsf{Sys})$ no nominal has more than one positive occurrence.

(SDNS2) In every consequent ψ of a sequent $\mathbf{j} \Rightarrow \psi$ of Sys, every positive occurrence of a nominal is at most in the scope of $\wedge$ and $\Diamond$ (i.e. not in the scope of any $\vee$, $\Box$, $\Diamond^{-1}$ or $\Box^{-1}$).

(SDNS3) In every consequent ψ of a sequent $\mathbf{j} \Rightarrow \psi$ of Sys,

- **(SDNS3.1)** every positive occurrence of a propositional variable is at most in the scope of $\wedge$, $\Diamond$ and $\Box$ (i.e. not in the scope of any $\vee$, $\Diamond^{-1}$ or $\Box^{-1}$), and
- **(SDNS3.2)** no positive occurrence of a propositional variable is in the scope of a $\Diamond$ which is in the scope of a $\Box$.

Lemma 5.2.28 *Let Sys be a SDNS. Then*

1. *any propositional variable which has a positive occurrence in Sys, can be eliminated from Sys by the application of the $\Diamond$, $\Box$, $\wedge$, and Ackermann-rules, yielding an SDNS Sys', and*

2. *any nominal $\mathbf{k}$ which has a positive occurrence in Sys, can be eliminated from Sys by the application of the $\Diamond$ and $\wedge$-rules as well as the Ackermann-rule for nominals, yielding an SDNS Sys'.*

Proof. The proof of (1) is essentially the same as that of lemma 2.6.2, except that we now also have to verify that conditions (SDNS1) and (SDNS2) are preserved. For the sake of (SDNS1), it is sufficient to show that the application of the $\Diamond$, $\Box$, $\wedge$, and Ackermann-rules does not increase the number of positive occurrences of a nominal in an SDNS. This is clear, even in arbitrary systems, for the $\wedge$ and $\Box$-rules, and also for the $\Diamond$-rule, as the positive nominal occurrences introduced by it are always new in the system. As far as the Ackermann-rule is concerned, we note the following: In an SDNS, even after the application of the $\wedge$,$\Box$ and $\Diamond$-rules, all nominal occurrences in antecedents are positive (hence negative in $\mathsf{Form}(\mathsf{Sys})$). Hence application of the Ackermann-rule introduces no new positive nominal occurrences in the system. From this fact the preservation of (SDNS2) also follows.

As for (2), since the single positive occurrence of $\mathbf{k}$ in Sys is at most in the scope $\Diamond$ and $\wedge$, we can apply the $\Diamond$ and $\wedge$-rules until the only sequent containing $\mathbf{k}$ positively is of the form $\mathbf{l} \Rightarrow \mathbf{k}$, for some nominal $\mathbf{l}$. From this system $\mathbf{k}$ is then eliminated by an application of the Ackermann-rule for nominals. That the application of the $\Diamond$ and $\wedge$-rules preserves (SDNS1), (SDNS2) and (SDNS3) follows as in the case for (1). Lastly it is clear that these properties are also preserved under the application of the Ackermann-rule for nominals. QED

Theorem 5.2.29 *SQEMA^{sd} succeeds on all nominalized Sahlqvist–van Benthem formulae.*

Proof. Let $\varphi \in \mathcal{L}^n$ be a nominalized Sahlqvist–van Benthem formula, and let φ_1 be the formula obtained by rewriting $\neg\varphi$ in negation normal form. In φ_1, no positive occurrence of a nominal is in the scope of a $\Box$, and moreover every two positive occurrence of a given nominal $\mathbf{j}$, are in a subformula of the form $\psi \vee \chi$, such that one occurrence is in ψ and the other is in χ. The formulae retains these properties also after the exhaustive distribution of $\Diamond$ and $\wedge$ over $\vee$. The resulting formula (i.e. the formula resulting from the preprocessing phase) is of the form $\bigvee_{i=1}^{n} \varphi_i$, where in each φ_i each disjunction occurrence is in the scope of $\Box$, and hence every nominal has at most one positive occurrence in any φ_i.

Now, for each φ_i, the system $\|\mathbf{i} \Rightarrow \varphi_i$, if not already an SDNS, may be transformed into an SDNS by applying the polarity switching rule. The theorem now follows by induction on the number of occurring propositional variables and input nominals with positive occurrences, using lemma 5.2.28. QED

Corollary 5.2.30 *All nominalized Sahlqvist–van Benthem formulae are sd-persistent.*

Corollary 5.2.31 *For any set Σ of nominalized Sahlqvist–van Benthem formulae, the logic $\mathbf{K}^n \oplus \Sigma$ is strongly sound and complete with respect to its class of Kripke frames.*

In subsection 5.3.3 we will introduce a more general class, which generalizes the monadic inductive formulae (definition 1.3.16) to $\mathcal{L}^{n,u}$, and hence also to $\mathcal{L}^n$.

5.3 The universal modality and satisfaction operator

Hybrid languages usually include, apart from nominals, either the universal modality or satisfaction operators ($@_\mathbf{i}$) to empower the nominals. In this section we consider the hybrid language $\mathcal{L}^{n,u}$, which extends the basic modal language $\mathcal{L}$ with nominals and the universal modality $[u]$. The universal modality is systematically studied in [GP92]. As already noted, the @-operator can be encoded with the universal modality in two ways, viz. $@_{\mathbf{j}}\varphi \equiv [\mathsf{u}][\mathbf{j} \to \varphi]$ or $@_{\mathbf{j}}\varphi \equiv \langle \mathsf{u} \rangle [\mathbf{j} \wedge \varphi]$.

In this section we wish to extend the algorithmic correspondence and completeness results of the previous section to $\mathcal{L}^{n,u}$. This will done by adding an additional preprocessing phase to SQEMA^{sd}, in which $\mathcal{L}^{n,u}$-formulae are 'flattened', by bringing all universal diamonds and boxes out under the scope of any other modal operators. The so extended algorithm will then enable us to obtain a new class of elementary and sd-persistent $\mathcal{L}^{n,u}$-formulae.

5.3.1 The language $\mathcal{L}^{n,u}$

Before introducing extensions of the algorithmic correspondence and completeness results of the previous, we consider a normal form for the language $\mathcal{L}^{n,u}$ and make some observations regarding correspondence result for this language.

We will refer to $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ as *universal modal operators* or simply *universal modalities*, while $\langle \mathsf{u} \rangle$ will be called the *universal diamond* and $[\mathsf{u}]$ will be called the *universal box*. To distinguish them from the universal modal operators, $\Diamond$ and $\Box$ will sometimes be referred to as *basic modal operators* or *basic modalities*.

In [GP92] it is shown that every $\mathcal{L}^{u}$-formula is equivalent to formulae in certain conjunctive and disjunctive normal forms, in which no $\langle \mathsf{u} \rangle$ or $[\mathsf{u}]$ occurs in the scope of any other modality, universal or otherwise. This latter property is of importance for our purposes. If $\varphi, \psi \in \mathcal{L}^{n,u}$, and $(\mathsf{u}) \in \{\langle \mathsf{u} \rangle, [\mathsf{u}]\}$, then the following equivalences are easy to verify:

(UM1) $\Box(\varphi \wedge (\mathsf{u})\psi) \equiv_{\text{sem}} \Box\varphi \wedge (\Box\bot \vee (\mathsf{u})\psi)$.

(UM2) $\Box(\varphi \vee (\mathsf{u})\psi) \equiv_{\text{sem}} \Box\varphi \vee (\mathsf{u})\psi$.

(UM3) $\Diamond(\varphi \vee (\mathsf{u})\psi) \equiv_{\text{sem}} \Diamond\varphi \vee (\Diamond\top \wedge (\mathsf{u})\psi)$.

(UM4) $\Diamond(\varphi \wedge (\mathsf{u})\psi) \equiv_{\text{sem}} \Diamond\varphi \wedge (\mathsf{u})\psi$.

Also, for any $(\mathsf{u})_1, (\mathsf{u})_2 \in \{\langle \mathsf{u} \rangle, [\mathsf{u}]\}$, we have

(UM5) $(\mathsf{u})_1(\varphi \wedge (\mathsf{u})_2\psi) \equiv_{\text{sem}} (\mathsf{u})_1\varphi \wedge (\mathsf{u})_2\psi$, and

(UM6) $(\mathsf{u})_1(\varphi \vee (\mathsf{u})_2\psi) \equiv_{\text{sem}} (\mathsf{u})_1\varphi \vee (\mathsf{u})_2\psi$.

A special cases of both (UM1) and (UM2) is $\Box(\mathsf{u})\psi \equiv_{\text{sem}} \Box\bot \vee (\mathsf{u})\psi$. Similarly, as a special case of both (UM3) and (UM4) we have $\Diamond(\mathsf{u})\psi \equiv_{\text{sem}} \Diamond\top \wedge (\mathsf{u})\psi$.

Definition 5.3.1 A $\mathcal{L}^{n,u}$-formula φ is a *flattened formula* if it contains no occurrence of universal modal operator which is in the scope of another modal operator, whether basic or universal.

We can flatten any $\mathcal{L}^{n,u}$-formula by the exhaustive application of the equivalences (UM1) to (UM6). We will refer to this procedure as *flattening*.

Proposition 5.3.2 ([GP92]) *Any $\mathcal{L}^{n,u}$-formula is semantically equivalent to a flattened formula.*

It [vB06], van Benthem considers so called *scattered versions* of modal formulae, which are obtained by assigning different indices for accessibility relations to occurrences of modal operators in a formula. It is pointed out that the formulae in the well known syntactic classes (e.g. Sahlqvist formulae) maintain their first-order definability under (partial) scattering. It is also shown that first-order definability may be lost under scattering, particularly, $(\Box p \to \Box\Box p) \wedge (\Box\Diamond \to \Diamond\Box p)$ is first order-definable ([vB76]) but the partially scattered version $(\Box_1 p \to \Box_1\Box_1 p) \wedge (\Box_2\Diamond_2 \to \Diamond_2\Box_2 p)$ is not. Now these remarks on scattering of course remain true if, rather than inserting indices, we replace some occurrences of $\Diamond$ and $\Box$ with $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$, respectively. The next example illustrates that first-order definability and persistence may sometimes, but certainly not always, be gained by the replacement of some basic modal operators with universal ones.

Example 5.3.3 Consider the formula $\Diamond\Box\neg p \vee \Diamond\Box p$, which is the McKinsey formula rewritten in negation normal form, which we know to be neither first-order definable ([vB83]) nor canonical ([Gol91]). Replacing the basic boxes with universal ones, we obtain $\Diamond[\mathsf{u}]\neg p \vee \Diamond[\mathsf{u}]p$. This formula is true at a point w in a Kripke frame $\mathfrak{F}$ if and only if $\mathfrak{F} \models Rxx \wedge \forall y \forall z(y = z)[x := w]$, i.e. it is true only in the Kripke frame that consist of a single, reflexive point. Since this is also the case for differentiated general frames, it follows that this formula is also (locally) d-persistent.

What happens when we replace the diamonds rather than the boxes, and obtain the formula $\langle \mathsf{u} \rangle\Box\neg p \vee \langle \mathsf{u} \rangle\Box p$? This formula is not globally (and hence not locally) first-order definable. To see this, we use a method similar to that used by van Benthem in [vB83] to show that the McKinsey axiom is not first-order definable. Consider the Kripke frame $\mathfrak{F}$ containing, for each $i \in \omega$, a pair of points v_i^0 and v_i^1 such that $Rv_i^0v_i^0$, $Rv_i^1v_i^1$, $Rv_i^0v_i^1$ and $Rv_i^1v_i^0$. Further, for each function $\rho : \omega \to \{0,1\}$ there is an irreflexive point z_ρ such that, for each $i \in \omega$, $Rz_\rho v_i^{\rho(i)}$. Now every point in $\mathfrak{F}$ validates $\langle \mathsf{u} \rangle\Box\neg p \vee \langle \mathsf{u} \rangle\Box p$.

Note that $\mathfrak{F}$ is uncountable. Now, by the downwards Skolem-Löwenheim theorem, there exists a countable, elementary substructure $\mathfrak{F}'$ of $\mathfrak{F}$ containing the subframe generated by the points in $\{v_i^j \mid i \in \omega, j = 0, 1\}$. Since there are uncountably many functions $\rho : \omega\{0,1\}$, there is a function $\rho_0 : \omega\{0,1\}$ such that $z_{\rho_0} \notin \mathfrak{F}'$. Call two functions $\rho_1, \rho_2 : \omega \to \{0,1\}$ *complementary* if $\rho_1(i) \neq \rho_2(i)$ for *all* $i \in \omega$. Complements are unique. We also call points z_{ρ_1} and z_{ρ_2} complementary if ρ_1 and ρ_2 are complementary. We can express the existence of a point complimentary to a point z in the frame using the first-order formula $\exists y(\neg Ryy \wedge \forall u(Ryu \leftrightarrow \neg Rzu))$. Hence, since $\mathfrak{F}'$ is an elementary substructure of $\mathfrak{F}$, it follows that $\mathfrak{F}'$ is closed under complementary points, i.e. $z_{\rho_1} \in \mathfrak{F}'$ if and only if $z_{\rho_2} \in \mathfrak{F}'$ where ρ_1 and ρ_2 are complementary. So, neither z_{ρ_0} nor its complementary point are in $\mathfrak{F}$. Evaluating p to the subset $\{v_i^{\rho_0(i)} \mid i \in \omega\}$ refutes $\langle \mathsf{u} \rangle\Box\neg p \vee \langle \mathsf{u} \rangle\Box p$ of $\mathfrak{F}$, as under this valuation every point has both a p and a non-p successor.

This example may also be construed as showing that we may not close the set of elementary $\mathcal{L}^{n,u}$-formulae under $\langle \mathsf{u} \rangle$, just like the set of elementary $\mathcal{L}$-formulae is not closed under $\Diamond$ (see [vB83]). $\lhd$

5.3.2 Extending SQEMA for the universal modality

How shall we handle the universal modality algorithmically? Of course, there is no impediment to feeding SQEMA^{sd} formulae from $\mathcal{L}^{n,u}$ as input, as $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ are, after all, a diamond and a box satisfying everything satisfied by $\Diamond$ and $\Box$, respectively. In other words all rules of SQEMA^{sd} may be applied to $\mathcal{L}^{n,u}$ and the algorithm will remain sound. By theorem 5.2.3 the accompanying completeness-via-persistence result will even hold.

But, of course, merely treating $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ like ordinary modalities ignores the special characteristics of a modality which must always be interpreted using the universal relation. Indeed, as we saw in example 5.3.3, the formula $\Diamond[\mathsf{u}]\neg p \vee \Diamond[\mathsf{u}]p$ is first-order definable. Yet SQEMA^{sd} will fail on it, since the algorithm will tret it exactly like the McKinsey formula. We would like our algorithm to at least be able to reduce such formulae.

We propose to incorporate all special treatment of the universal modality in an additional preprocessing phase, which will convert the (unnegated) input formula into negation normal form and then flatten it using (UM1) to (UM6). This formula, is then fed further through the algorithm, which need not be changed in any way, as usual. Universal diamonds and boxes are further treated like basic diamonds and boxes. Formally:

Definition 5.3.4 Let SQEMA^{u} be the algorithm obtained by prefixing SQEMA^{sd} with a flattening procedure, which applies equivalences (UM1) to (UM6) to flatten $\mathcal{L}^{n,u}$ input formulae after conversion into negation normal form. The resulting formulae are then passed to SQEMA^{sd} where $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ are treated like ordinary diamonds and boxes.

One might wish to have an extension of SQEMA which handles input formulae from $\mathcal{L}^{u}$, i.e., the basic modal language enriched with the universal modality but without nominals. However, notice that SQEMA and SQEMA^{sd} are equivalent as far as $\mathcal{L}$-input formulae are concerned. Hence whether we prefix SQEMA or SQEMA^{sd} with a flattening procedure for the universal modality, the resulting algorithm will treat $\mathcal{L}^{u}$-formulae in the same way.

Example 5.3.5 Consider the formula $\Diamond[\mathsf{u}]\neg p \vee \Diamond[\mathsf{u}]p$ form example 5.3.3. SQEMA^{u} first flattens this formula to become $(\Diamond\top \wedge [\mathsf{u}]\neg p) \vee (\Diamond\top \wedge [\mathsf{u}]p)$. Next this is given to SQEMA^{sd} as input.

Phase 1: Negation yields $(\Box\bot \vee \langle \mathsf{u} \rangle p) \wedge (\Box\bot \vee \langle \mathsf{u} \rangle \neg p)$, which after distribution becomes

$$(\Box\bot \wedge \Box\bot) \vee (\Box\bot \wedge \langle \mathsf{u} \rangle \neg p) \vee (\langle \mathsf{u} \rangle p \wedge \Box\bot) \vee (\langle \mathsf{u} \rangle p \wedge \langle \mathsf{u} \rangle \neg p).$$

Phase 2: There are four initial systems, namely $\|\mathbf{i} \Rightarrow (\Box\bot \wedge \Box\bot)$, $\|\mathbf{i} \Rightarrow (\Box\bot \wedge \langle \mathsf{u} \rangle \neg p)$, $\|\mathbf{i} \Rightarrow (\langle \mathsf{u} \rangle p \wedge \Box\bot)$ and $\mathbf{i} \Rightarrow (\langle \mathsf{u} \rangle p \wedge \langle \mathsf{u} \rangle \neg p)$. The first system contains no propositional variables, while the second and third are negative and positive in p, and hence p is eliminated by the substitution of $\bot$ and $\top$, respectively. In the case of the fourth system the $\wedge$-rule is applied to obtain

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle p \\ \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle \neg p \end{array} \right. ,$$

which the $\Diamond$-rule transforms into

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle \mathbf{j} \\ \mathbf{j} \Rightarrow p \\ \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle \neg p \end{array} \right. .$$

The Ackermann-rule is now applicable yielding the pure system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle \mathbf{j} \\ \mathbf{i} \Rightarrow \langle \mathsf{u} \rangle \neg \mathbf{j} \end{array} \right. .$$

We obtain the formula

$$(\neg \mathbf{i} \vee (\Box \bot \wedge \Box \bot)) \vee (\neg \mathbf{i} \vee (\Box \bot \wedge \langle \mathsf{u} \rangle \top)) \vee (\neg \mathbf{i} \vee (\langle \mathsf{u} \rangle \top \wedge \Box \bot)) \vee ((\neg \mathbf{i} \vee \langle \mathsf{u} \rangle \mathbf{j}) \wedge (\neg \mathbf{i} \vee \langle \mathsf{u} \rangle \neg \mathbf{j})).$$

as $\mathsf{pure}(\varphi)$. Negating and simplifying somewhat yields

$$\mathbf{i} \wedge (\Diamond \top \wedge ([\mathsf{u}] \neg \mathbf{j} \vee [\mathsf{u}] \mathbf{j}))$$

which locally defines the property $\models Rxx \wedge \forall y \forall z (y = z)$, as expected.

$\lhd$

SQEMA^u is clearly sound with respect to Kripke frames. Indeed, this follows from the soundness of SQEMA^{sd} and the validity of the equivalences (UM1) to (UM6). Moreover SQEMA^u is sound with respect to strongly descriptive frames. For, notice that the presence of the universal modality in the language makes no additional demands on the algebras of admissible sets of general frames over which this language is interpreted. Indeed, in any model $\mathcal{M} = (W, R, V)$, the extension of any formula of the form $\langle \mathsf{u} \rangle \varphi$ or $[\mathsf{u}]\varphi$ is either W or $\emptyset$. Hence both $[\mathsf{u}]$ and $\langle \mathsf{u} \rangle$ are both open and closed operators on strongly descriptive frames. Using this fact, it is not difficult to verify (we leave this to the patient reader) that theorem 5.2.20 can be generalized to SQEMA^u. Hence we have

Theorem 5.3.6 *Every $\mathcal{L}^{n,u}$-formula on which SQEMA^u succeeds is locally elementary and globally sd-persistent.*

Corollary 5.3.7 *The logic $\mathbf{K}^{n,u} \oplus \Sigma$ is strongly sound and complete with respect to its Kripke frames whenever SQEMA^u succeeds on all members of Σ.*

In closing this subsection, we remark that it is possible to add some auxiliary rules to SQEMA^{sd} which are tailored for the universal modality. For example, axiomatizations of logics with the universal modality usually contain the so-called *inclusion axiom*, $[\mathsf{u}]p \to \Box p$. Indeed, it is clear that $[\mathsf{u}]p \wedge \Box p \equiv_{\text{sem}} [\mathsf{u}]p$. This latter equivalence can be useful for the simplification of formulae, and such equivalences are in fact employed in an extension[2] of the implementation of SQEMA by Dimiter Georgiev ([Geo06]).

[2]Not yet available online at the time of writing.

5.3.3 Two syntactic classes

In this subsection we use SQEMA^u to prove the elementarity and sd-persistence of extended classes of Sahlqvist–van Benthem and inductive formulae.

The universalized Sahlqvist–van Benthem formulae

Extend the definition of the nominalized Sahlqvist–van Benthem formulae as follows:

Definition 5.3.8 An $\mathcal{L}^{n,u}$-formula is a *universalized Sahlqvist–van Benthem formula* if it is in negation normal form and satisfies the conditions:

(USB1) For every occurring propositional variable p, either

- **(USB1.1)** there is no positive occurrence of p in a subformula $\psi \wedge \chi$ or $\Box\psi$ which is in the scope of a $\Diamond$ or $\langle \mathsf{u} \rangle$, or
- **(USB1.2)** there is no negative occurrence of p in a subformula $\psi \wedge \chi$ or $\Box\psi$ which is in the scope of a $\Diamond$ or $\langle \mathsf{u} \rangle$.

(USB2) No negative nominal occurrence in φ is in the scope of a $\Diamond$ or $\langle \mathsf{u} \rangle$.

(USB3) Every two negative occurrences of a given nominal $\mathbf{j}$ are in a subformula of the form $\psi \wedge \chi$ such that one occurrence is in ψ and the other is in χ.

Note that the definition imposes the same constraints on $\langle \mathsf{u} \rangle$ as on $\Diamond$, but that it makes no restrictions whatsoever on the occurrences of $[\mathsf{u}]$.

Example 5.3.9 The formula $\Diamond[\mathsf{u}]\neg p \vee \Diamond[\mathsf{u}]p$ from examples 5.3.3 and 5.3.5 is a universalized Sahlqvist–van Benthem formula. The formula $\langle \mathsf{u} \rangle\Box\neg p \vee \langle \mathsf{u} \rangle\Box p$, also from example 5.3.3, is not a universalized Sahlqvist–van Benthem formula, as it violates condition (USB1). Similarly, the formula $\langle \mathsf{u} \rangle\Box\neg\mathbf{j} \vee \langle \mathsf{u} \rangle\Box\mathbf{j}$ violates condition (USB2). $\lhd$

Theorem 5.3.10 *SQEMA^u succeeds on every universalized Sahlqvist–van Benthem formula.*

Proof. (Sketch) The class of universalized Sahlqvist–van Benthem formulae is closed under the application (from left to right) of the equivalences (UM1) to (UM6) used to flatten formulae. In flattened universalized Sahlqvist–van Benthem formulae all constraints imposed by definition 5.3.8 on occurrences of $\Box$ are also satisfied by all occurrences of $[\mathsf{u}]$. Hence these formulae may be seen as bi-modal nominalized Sahlqvist–van Benthem formulae (definition 5.2.25), for which the proofs of lemma 5.2.28 and theorem 5.2.29 also go through. QED

Corollary 5.3.11 *All universalized Sahlqvist–van Benthem formulae are locally elementary and globally sd-persistent, and hence axiomatize extensions of $\mathbf{K}^{n,u}$ which are strongly sound and complete with respect to elementary classes of Kripke frames.*

The universalized inductive formulae

We now introduce a more general class, subsuming the universalized Sahlqvist–van Benthem-formulae, by generalizing the monadic inductive formulae (definition 1.3.16) to $\mathcal{L}^{n,u}$. In [GV01], Goranko and Vakarelov extend the definition of the inductive formulae to $\mathcal{L}^n$, essentially by allowing arbitrary occurrences of nominals. Although the members of this class are (locally) elementary, the definition is too liberal to ensure a suitable persistence property. Our aim is to obtain a class of elementary and sd-persistent formulae, and we will thus have to be more circumspect in the occurrences of nominals which we allow. As regards the universal modality, one could opt to treat $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ like $\Diamond$ and $\Box$, respectively. However, taking the special properties of the universal modality into account, we are able to alow some additional occurrences which do not correspond to occurrences of $\Diamond$ and $\Box$.

Recall that proposition variables and nominals together are referred to as *atoms*. We say an $\mathcal{L}^n$-formula is *absolutely positive* (*absolutely negative*) if all occurrences of atoms in it are positive (negative).

Definition 5.3.12 Let $\sharp$ be a symbol not belonging to $\mathcal{L}^{n,u}$. Then a *universalized box-form of* $\sharp$ in $\mathcal{L}^{n,u}$ is defined recursively as follows:

1. $\sharp$ is a universalized box-form of $\sharp$;
2. If $\mathbf{B}(\sharp)$ is a universalized box-form of $\sharp$, then $\Box\mathbf{B}(\sharp)$, $[\mathsf{u}]\mathbf{B}(\sharp)$ and $\langle \mathsf{u} \rangle\mathbf{B}(\sharp)$ are universalized box-forms of $\sharp$;
3. If $\mathbf{B}(\sharp)$ is a universalized box-form of $\sharp$ and A is an absolutely positive $\mathcal{L}^{n,u}$-formula, then $A \to \mathbf{B}(\sharp)$ is a universalized box-form of $\sharp$.

Definition 5.3.13 By substituting a propositional variable p for $\sharp$ in a universalized box-form $\mathbf{B}(\sharp)$ we obtain a *universalized box-formula of* p, namely $\mathbf{B}(p)$. The last occurrence of the propositional variable p is the *head* of $\mathbf{B}(p)$. Every occurrence of a propositional variable in a universalized box formula other than the head is called *inessential* there.

Definition 5.3.14 A *universalized pre-regular formula* is an $\mathcal{L}^{n,u}$-formula built up from absolutely positive formulae, negated nominals and negated universalized box-formulae by applying conjunctions, disjunctions, $\Box$'s and $[\mathsf{u}]$'s, in such a way that no nominal has more than one negative occurrence.

Definition 5.3.15 The *dependency digraph* of a set $\mathcal{B} = \{\mathbf{B}_1(p_1), \ldots, \mathbf{B}_n(p_n)\}$ of universalized box-formulae is a digraph $G_{\mathcal{B}} = \langle V, E \rangle$ where $V = \{p_1, \ldots, p_n\}$ is the set of heads in $\mathcal{B}$, and the edge set E is such that $p_i E p_j$ iff p_i occurs inessentially in a universalized box-formula from $\mathcal{B}$ with a head p_j. A digraph is *acyclic* if it does not contain oriented cycles or loops. We will also talk about the dependency digraph of a formula, when we mean the dependency digraph of the set of universalized box-formulae that occur as subformulae of that formula.

Definition 5.3.16 A *universalized pre-inductive formula (UPIF)* is a universalized pre-regular formula with an acyclic dependency digraph. A *universalized inductive formula (UIF)* is any formula built up from UPIF's by applying conjunctions, $\Box$'s and $[\mathsf{u}]$'s.

Example 5.3.17 The formula

$$\neg p \vee \neg\Box[\mathsf{u}](\Diamond(p \wedge \Diamond\mathbf{j}) \rightarrow \langle\mathsf{u}\rangle q) \vee [\mathsf{u}](\Diamond\Box\Box(q \vee \mathbf{j}) \vee [\mathsf{u}]\neg\mathbf{j}),$$

is a universalized inductive formula, while the formulae

$$\neg p \vee \neg\Box[\mathsf{u}](\Diamond(p \wedge \Diamond\mathbf{j}) \rightarrow \langle\mathsf{u}\rangle q) \vee [\mathsf{u}](\Diamond\Box\Box(q \vee \mathbf{j}) \vee \langle\mathsf{u}\rangle\neg\mathbf{j}),$$

and

$$\neg p \vee \neg\mathbf{j} \vee \neg\Box(\Diamond(p \wedge \Diamond\mathbf{j}) \rightarrow \Box q) \vee \Diamond\Box\Box(q \vee \mathbf{j}) \vee [\mathsf{u}]\neg\mathbf{j},$$

are not. Indeed, it is impossible to construct the latter two formulae in accordance with definition 5.3.16, as the first contains a negative occurrence of $\mathbf{j}$ in the scope of a $\langle\mathsf{u}\rangle$, while the second contains two occurrences of $\neg\mathbf{j}$ not separated by any conjunction. $\lhd$

Proposition 5.3.18 *Every universalized Sahlqvist–van Benthem-formula is locally equivalent to a universalized inductive formula.*

Proof. Let φ be a universalized Sahlqvist–van Benthem-formula, and let φ_1 be obtained from φ by switching the polarity of propositional variables in such a way that condition (USB1.2) holds with respect to every occurring propositional variable. Let φ_2 be the formula obtained from φ_1 by distributing boxes and disjunctions over conjunctions not in the scope of any $\Diamond$ or $\langle\mathsf{u}\rangle$, as much as possible. Hence φ_2 is of the form $\bigwedge \psi_i$ where in each ψ_i is a universalized Sahlqvist–van Benthem-formula, such that, in it:

1. each nominal has at most one negative occurrence,

2. no negative occurrence of a propositional variable is in the scope of a $\Box$ or $\wedge$ which is in the scope of a $\Diamond$ or $\langle\mathsf{u}\rangle$.

In each conjunct ψ_i of φ_2, distribute $\Diamond$'s and $\langle\mathsf{u}\rangle$'s over disjunctions as much as possible to obtain φ_i'. Call a negated propositional variable preceded by finitely (possibly 0) many $[\mathsf{u}]$'s, $\Diamond$'s and/or $\langle\mathsf{u}\rangle$'s a *prefixed negative propositional variable*. Now, each ψ_i' is built up from absolutely positive formulae, prefixed negative propositional variables, and negated nominals using $\vee$, $\Box$ and $[\mathsf{u}]$. But, by pulling out the negation in a diamonded negative propositional variable $\circ_1 \cdots \circ_n \neg p$, with $\circ_1, \ldots, \circ_n \in \{\Diamond, \langle\mathsf{u}\rangle, [\mathsf{u}]\}$, we obtain a formula $\neg \circ_1' \cdots \circ_n' p$, with $\circ_1', \ldots, \circ_n' \in \{\Box, \langle\mathsf{u}\rangle, [\mathsf{u}]\}$, which is a negated universalized box formula of p. Hence every ψ_i' is a universalized pre-inductive formula with empty dependency digraph. We conclude that φ is locally equivalent to a universalized inductive formula. QED

Now we will show that SQEMA^u succeeds on all universalized inductive formulae, thus proving the elementarity and sd-persistence of these formulae. To this aim we will exploit the resemblance of these formulae to monadic inductive formulae (from $\mathcal{L}$). The feature which distinguishes the universalized inductive formulae most sharply from the inductive formulae, is probably the use of $\langle\mathsf{u}\rangle$ in the construction of box-formulae. The following definition removes this feature:

Definition 5.3.19 We shall call a box form constructed without using the clause for $\langle \mathsf{u} \rangle$ in definition 5.3.12 a *semi-universalized box form.* By substituting a propositional variable p for $\sharp$ in a semi-universalized box form $\mathbf{B}(\sharp)$ we obtain a *semi-universalized box formula* $\mathbf{B}(p)$ of p. The class of *semi-universalized pre-regular formulae* is obtained by replacing 'universalized box formulae' with 'semi-universalized box formulae' in definition 5.3.14. The *semi-universalized pre-inductive* (SUPIF) and *semi-universalized inductive formulae* (SUIF) are defined similarly.

Example 5.3.20 $\Box((p \wedge q \wedge \langle \mathsf{u} \rangle \mathbf{j}) \to [\mathsf{u}](\mathbf{j} \to p))$ is a semi-universalized box formula, but $\Box((p \wedge q \wedge \langle \mathsf{u} \rangle \mathbf{j}) \to \langle \mathsf{u} \rangle(\mathbf{j} \to p))$ and $\Box\langle \mathsf{u} \rangle((p \wedge q \wedge \langle \mathsf{u} \rangle \mathbf{j}) \to [\mathsf{u}](\mathbf{j} \to p))$ are not, as both contain occurrences of $\langle \mathsf{u} \rangle$ which are not within absolutely positive subformulae. $\lhd$

Remark 5.3.21 By a trivial (universalized or semi-universalized) box formula we mean a (universalized or semi-universalized) box formula obtained as an instance of the basic box form $\sharp$. Note that, if we were to allow nominals as the heads of box-formulae, then the negative nominal occurrences in a universalized inductive formula could be regarded as the negations of trivial box formulae with those nominals as heads. Thus, a semi-universalized pre-inductive formula can be regarded as a inductive formula (in the sense of definition 1.3.16) in which

1. $\langle \mathsf{u} \rangle$ and $[\mathsf{u}]$ are also allowed to occur, and are treated in the same way as $\Diamond$ and $\Box$, respectively;
2. nominals are also allowed to occur, and are treated in the same way as propositional variables, except for the fact that
   (a) the only box-formulae that may have nominals as heads are trivial box formulae, and
   (b) for every nominal $\mathbf{j}$, at most one occurring box formula may have $\mathbf{j}$ as head.

Theorem 5.3.22 *SQEMA^u succeeds on all universalized inductive formulae.*

Proof.(Sketch) We will use the following claim:

Claim 1 Applying the flattening rules (UM1) to (UM6) to a UIF rewritten in negation normal form, yields the negation normal form of a SUIF.

Proof of Claim It sufficed to consider the effect of the rules (UM1) to (UM6) on negated universalized box-formulae. Note that the negation of a universalized box formula, rewritten in negation normal form, has the form $\overline{\circ}_1(A_1 \wedge \overline{\circ}_2(A_2 \wedge \cdots \overline{\circ}_n(A_n \wedge \neg p) \cdots))$ where each A_i is an absolutely positive $\mathcal{L}^{n,u}$-formula, and each $\overline{\circ}_i$ is a finite sequence of elements from $\{\Diamond, [\mathsf{u}], \langle \mathsf{u} \rangle\}$. Exhaustive application of the rules (UM1) to (UM6) transforms this formula into one of the form $\bigwedge \psi_i$, where each ψ_i is either

1. an absolutely positive $\mathcal{L}^{n,u}$-formula,

2. a formula of the form $\circ_0\underline{\Diamond}_1(B_1 \wedge \underline{\Diamond}_2(B_2 \wedge \cdots \underline{\Diamond}_m(B_m \wedge \neg p)\cdots))$, or

3. a formula of the form $\underline{\Diamond}_1(B_1 \wedge \underline{\Diamond}_2(B_2 \wedge \cdots \underline{\Diamond}_m(B_m \wedge \neg p)\cdots))$

where each B_i is an absolutely positive $\mathcal{L}^{n,u}$-formula, $\circ_0 \in \{\langle \mathsf{u}\rangle, [\mathsf{u}]\}$ and each $\underline{\Diamond}_i$ is finite sequence of $\Diamond$'s. This formula is of the desired shape. ◀

It follows from claim 1, that the flattening phase of SQEMA^u transforms a universalized inductive formula φ into (the negation normal forms of) a SUIF. Hence, the preprocessing phase (by negating and distributing $\wedge$, $\Diamond$ and $\langle \mathsf{u}\rangle$ over $\vee$) transforms this formula into one of the form $\bigvee \varphi_i$ where each φ_i is the negation normal form of a negated SUPIF.

Now, by remark 5.3.21, a SUPIF can be regarded essentially as an inductive formula. Hence, if we extend the dependency digraph to include nominals as vertices, the proofs of lemma 2.6.7 and theorem 2.6.8 can now be readily generalized, by noting the following: Since the only 'box formulae' with nominals as heads are trivial, all nominals will be isolated points in the dependency digraph, and may hence be inserted at the beginning of some linear order extending the order induced by this digraph. Moreover, the positive nominal occurrences corresponding to these heads are unique in each initial system (corresponding to a SUPIF), and occur at most in the scope of $\Diamond$, $\langle \mathsf{u}\rangle$ and $\wedge$. It follows that these occurrences can be eliminated by the application of the $\wedge$ and $\Diamond$-rules as well as the Ackermann-rule for nominals. Propositional variables are next eliminated as in the proof of lemma 2.6.7. QED

5.3.4 The satisfaction operator

As we have already remarked, every $\mathcal{L}^{n,@}$-formula can be translated into a semantically equivalent $\mathcal{L}^{n,u}$ formula. Via this translation we can handle any $\mathcal{L}^{n,@}$-input formula with SQEMA^u. Accordingly we will not introduce a version SQEMA specifically for $\mathcal{L}^{n,@}$, but confine ourselves to some remarks on the use of SQEMA^u for finding first-order correspondents for $\mathcal{L}^{n,@}$-formulae and proving their sd-persistence. Recall that, by theorem 5.2.3, we may deduce the completeness of logics axiomatized with sd-persistent formulae also in the case of $\mathcal{L}^{n,@}$-formulae.

Since we can translate $@_\mathbf{j}\varphi$ either as $[\mathsf{u}](\mathbf{j} \to \varphi)$ or as $\langle \mathsf{u}\rangle(\mathbf{j} \wedge \varphi)$, the question arises — is one form preferable to the other when our aim is to feed the result to SQEMA^u? Would one form be more likely to lead to a successful execution? The answer is not clear cut. If we regard the universalized Sahlqvist–van Benthem formulae (definition 5.3.8) as typical of the formulae on which SQEMA^u succeeds, we see that we are presented with two competing constraints. Translating $@_\mathbf{j}\varphi$ as $[\mathsf{u}](\mathbf{j} \to \varphi)$ introduces a negative nominal occurrence. These occurrences are limited by (USB2) and (USB3). On the other hand, translating $@_\mathbf{j}\varphi$ as $\langle \mathsf{u}\rangle(\mathbf{j} \wedge \varphi)$ places φ in the scope of a $\wedge$ and a $\langle \mathsf{u}\rangle$ — a combination which could very likely cause the resulting formula to violate condition (USB1). So no blanket policy can be formulated here.

Example 5.3.23 Consider the formula $@_\mathbf{j}\Diamond\mathbf{k} \vee @_\mathbf{j}\mathbf{k} \vee @_\mathbf{k}\Diamond\mathbf{j}$. This is a weaker (non-exclusive) version of trichotomy which is not $\mathcal{L}$-definable. Translating this formula as $[\mathsf{u}](\mathbf{j} \to \Diamond\mathbf{k}) \vee [\mathsf{u}](\mathbf{j} \to \mathbf{k}) \vee [\mathsf{u}](\mathbf{k} \to \Diamond\mathbf{j})$, and feeding it to SQEMA^u yields an in initial system

$$\left\|\begin{array}{l} \mathbf{i} \Rightarrow \langle \mathsf{u}\rangle(\mathbf{j} \wedge \Box\neg\mathbf{k}) \\ \mathbf{i} \Rightarrow \langle \mathsf{u}\rangle(\mathbf{j} \wedge \neg\mathbf{k}) \\ \mathbf{i} \Rightarrow \langle \mathsf{u}\rangle(\mathbf{k} \wedge \Box\neg\mathbf{j}) \end{array}\right. .$$

This system contains two positive occurrences of the input nominal $\mathbf{j}$. Hence SQEMA^u will fail to eliminate this nominal. Had we translated the original formula as $\langle \mathsf{u} \rangle(\mathbf{j} \wedge \Diamond\mathbf{k}) \vee \langle \mathsf{u} \rangle(\mathbf{j} \wedge \mathbf{k}) \vee \langle \mathsf{u} \rangle(\mathbf{k} \wedge \Diamond\mathbf{j})$, this would have yielded an initial system

$$\left\Vert \begin{array}{l} \mathbf{i} \Rightarrow [\mathsf{u}](\neg\mathbf{j} \vee \Box\neg\mathbf{k}) \\ \mathbf{i} \Rightarrow [\mathsf{u}](\neg\mathbf{j} \vee \neg\mathbf{k}) \\ \mathbf{i} \Rightarrow [\mathsf{u}](\neg\mathbf{k} \vee \Box\neg\mathbf{j}) \end{array} \right. ,$$

containing no propositional variables or positive occurrences of input nominals. Hence there is noting for SQEMA^u to do in this case, and the algorithm succeeds trivially. We can also conclude that $@_{\mathbf{j}}\Diamond\mathbf{k} \vee @_{\mathbf{j}}\mathbf{k} \vee @_{\mathbf{k}}\Diamond\mathbf{j}$ is sd-persistent. $\lhd$

Chapter 6

Semantic Extensions of SQEMA

Glancing over the proof of Ackermann's Lemma (lemma 2.1.4), one notices than something somewhat stronger is actually being proven. To be precise, instead of the negativity of the formula B w.r.t. the propositional variable p, the weaker, derivative property of B's downward monotonicity is actually used. Hence we immediately have the following, stronger result.

Lemma 6.0.1 *Let $A, B(p)$ be $\mathcal{L}^n_{r(\tau)}$-formulae such that the propositional variable p does not occur in A and $B(p)$ is downwards monotone in p. Then for any model $\mathcal{M}$, $\mathcal{M} \Vdash B(A)$ iff $\mathcal{M}' \Vdash (A \to p) \wedge B(p)$ for some model $\mathcal{M}'$ which may only differ from $\mathcal{M}$ on the valuation of p.*

This immediately suggests a stronger version of the Ackermann rule, viz. one that requires monotonicity rather than negativity in the propositional variable under consideration. But can we still effectively determine the applicability of such a rule? The answer is yes, for we have:

Lemma 6.0.2 *An $\mathcal{L}^n_{r(\tau)}$-formula $\varphi(p)$ is downwards monotone in p iff*

$$\Vdash \varphi(p) \to \varphi(p \wedge q)$$

where q is any variable not occurring in $\varphi(p)$.

Hence the question of the monotonicity of an $\mathcal{L}^r_n$-formula in a propositional variable can be effectively reduced to the question of the validity of a related $\mathcal{L}^r_n$-formula, a problem which is decidable and EXPTIME-complete (see [ABM00]). (By the way, note that testing validity is effectively reducible to testing monotonicity: $\Vdash \varphi$ iff $q \to \varphi$ is upwards monotone in q, where q is a variable not occurring in φ.)

In this chapter we explore some repercussions of this simple insight. In particular we develop 'semantic' versions of SQEMA. The word *semantic* is intended to be suggestive of the fact that we have exchanged the syntactic property of negative/positive polarity for its semantic correlate — monotonicity. Several results relating polarity monotonicity and polarity in (fragments of) the languages we are interested in will be proven further. These are so called 'Lyndon-type' theorems, in honour of Lyndon's famous theorem, proved in [Lyn59b], stating that a first-order formula is preserved under surjective homomorphisms if and only if it is equivalent to a positive formula. Lyndon interpolation theorems ([Lyn59a]) are closely related.

6.1 Two semantic extensions of SQEMA

When basing a transformation rule on lemma 6.0.1 we are faced with two choices, namely whether to substitute A directly for p in $B(p)$, or first to replace B with a 'syntactically correct' equivalent, viz. an equivalent which is negative in p and syntactically open. The first option is sometimes more efficient, but may cause the algorithm to fail later on. Moreover, it is not clear whether the formulae reducible by the algorithm when this option is chosen are always d-persistent. The second option, although computationally less efficient, allows us to prove the desired d-persistence result. We introduce two extensions of SQEMA, corresponding to these two options. Both versions of the algorithm are illustrated with some examples.

6.1.1 An extension without replacement

The most straightforward extension of the SQEMA-algorithm which takes advantage of this fact of the decidability of monotonicity, simply replaces the Ackermann rule with the following 'semantic' version. We will refer to this rule as the *Semantic Ackermann-rule without replacement*.

$$\text{The system} \quad \left\| \begin{array}{l} A_1 \Rightarrow p, \\ \vdots \\ A_n \Rightarrow p, \\ B_1(p), \\ \vdots \\ B_m(p), \end{array} \right. \quad \text{is replaced by} \quad \left\| \begin{array}{l} B_1((A_1 \vee \ldots \vee A_n)/p), \\ \vdots \\ B_m((A_1 \vee \ldots \vee A_n)/p). \end{array} \right.$$

where:

1. p does not occur in $A_1, \ldots, A_n$;
2. $\mathsf{Form}(B_1) \wedge \cdots \mathsf{Form}(\wedge B_m)$ is downwards monotone in p.

Two comments are in order:

1. Notice the requirement that the *conjunction* $B_1 \wedge \cdots \wedge B_m$, rather than the individual sequents be downwards monotone. Since monotonicity as a property is generally not preserved under taking subformulae, this ensures a wider applicability of the rule.

2. We also note that the sequents $A_1 \Rightarrow p, \ldots, A_n \Rightarrow p$ still have to have the right syntactic shape, so the rule in fact mixes syntactic and sematic requirements in its conditions of applicability.

The version of SQEMA obtained by replacing the ordinary Ackermann-rule with the semantic Ackermann-rule without replacement will be referred to as SSQEMA (for *Semantic* SQEMA). We illustrate SSQEMA with some examples:

Example 6.1.1 Consider the formula $\varphi = \neg(\Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q) \wedge \Box p \wedge \Box\Diamond\neg q)$. As the reader can check, SQEMA will fail on this formula. Applying the $\wedge$-rule to the resulting initial system we obtain:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q) \\ \mathbf{i} \Rightarrow \Box p \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

Note that this system, although it can be solved for q, cannot be solved for p, since neither the positive nor the negative occurrence of p in the first sequent can be isolated by the application of transformation rules. However, the first sequent is downwards monotone in p. Indeed, the formula $\Box\Diamond p \vee \Diamond\Diamond\neg p$ is easily seen to be semantically equivalent to $\Box\Diamond\top \vee \Diamond\Diamond\neg p$. Hence, if we apply the $\Box$-rule to obtain the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q) \\ \Diamond^{-1}\mathbf{i} \Rightarrow p \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. ,$$

we can apply the semantic Ackermann-rule to eliminate p:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box\Diamond\Diamond^{-1}\mathbf{i} \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i} \vee q) \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

Notice that the consequent of the first sequent is not syntactically open, since it contains both a positive occurrence of $\Diamond^{-1}$ and a positive nominal occurrence. Solving for q we obtain

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \wedge \neg(\Box\Diamond\Diamond^{-1}\mathbf{i} \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i}) \Rightarrow q \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

Now applying the Ackermann-rule we get

$$\| \ \mathbf{i} \Rightarrow \Box\Diamond(\Box^{-1}\neg\mathbf{i} \vee (\Box\Diamond\Diamond^{-1}\mathbf{i} \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i})) \ .$$

In this application of the Ackermann-rule the formula A in the Akcermann-equivalence $(A \to p) \wedge B(p)$ is not syntactically closed. This means that lemma 2.5.21 is not true for SSQEMA, and hence that we may not appeal directly to proposition 2.5.22 to conclude that this transformation preserves equivalence on descriptive frames. Of course, by noting the equivalence $\Box\Diamond p \vee \Diamond\Diamond\neg p \equiv_{\mathrm{sem}} \Box\Diamond\top \vee \Diamond\Diamond\neg p$, one sees that, in this case, A is semantically *equivalent* to a syntactically open formula, and hence equivalence on descriptive frames is indeed preserved. $\lhd$

Example 6.1.2 In the previous example, although the criteria of syntactic closedness and openness were violated in the last application of the Ackermann-rule, we were yet able to claim that, modulo semantic equivalence, these criteria were actually met. Here is an example which illustrates that this is not always the case.

Consider the input formula $\neg(\Box(\Box(p \vee q) \wedge \Box\neg p) \wedge \Box p \wedge \Box\Diamond\neg q)$. As it happens, SQEMA will succeed on this input formula, but some interesting things happen when SSQEMA is run on it: Applying the $\wedge$ and $\Box$-rules to the corresponding initial system we obtain

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box(p \vee q) \wedge \Box\neg p) \\ \Diamond^{-1}\mathbf{i} \Rightarrow p \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

The first sequent is downwards-monotone in p, as $\Box(\Box(p \vee q) \wedge \Box\neg p) \equiv_{\mathrm{sem}} \Box\Box(q \wedge \neg p)$. Applying the semantic Ackermann-rule yields

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box(\Diamond^{-1}\mathbf{i} \vee q) \wedge \Box\Box^{-1}\neg\mathbf{i}) \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

The variable q still remains to be eliminated. Applying the $\Box$, $\wedge$ and left-shift $\vee$-rules to the first sequent transforms the system into

$$\left\| \begin{array}{l} \Diamond^{-1}\Diamond^{-1}\mathbf{i} \wedge \Box^{-1}\neg\mathbf{i} \Rightarrow q \\ \Diamond^{-1}\mathbf{i} \Rightarrow \Box\Box^{-1}\neg\mathbf{i} \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

We may now apply the Ackermann-rule to eliminate q, but notice that $\Diamond^{-1}\Diamond^{-1}\mathbf{i} \wedge \Box^{-1}\neg\mathbf{i}$ is not syntactically closed, *nor is it equivalent to a syntactically closed formula* (this can easily be seen by constructing a suitable syntactically closed simulation, as introduced in section 6.2). So the argument used in example 6.1.1 to show the maintenance of equivalence on descriptive frames *does not work*. Of course, had we replaced $\Box(\Box(p \vee q) \wedge \Box\neg p)$ with its equivalent $\Box\Box(q \wedge \neg p)$ to begin with, the algorithm would still have succeeded and this problem would not have arisen. What happens here is that $\Box(\Box(p \vee q) \wedge \Box\neg p)$ (after substitution $\Box(\Box(\Diamond^{-1}\mathbf{i} \vee q) \wedge \Box\Box^{-1}\neg\mathbf{i})$) gets 'split up' by the application of transformation rules into subformulae which are no longer downwards monotone in p, and which, hence, no longer have equivalents which are negative in p. $\lhd$

The above example illustrates the main impediment that we encounter when trying to extend theorem 2.5.23 to show that all $\mathcal{L}$-formulae on which SSQEMA succeeds are d-persistent. Indeed, the proof of the latter theorem depends crucially on lemma 2.5.21 stating that the non-pure antecedents (consequents) of sequents during SQEMA-executions are syntactically closed (open). This property is lost (even modulo equivalence, as seen above) when we admit the semantic Ackermann rule.

Question 6.1.3 Are all $\mathcal{L}$-formulae on which SSQEMA succeeds d-persistent?

6.1.2 An extension with replacement

The problems with the syntactic shape of sequents (e.g. not being syntactically closed / open) encountered in examples 6.1.1 and 6.1.2 suggest a modification of the semantic Ackermann-rule. Specifically, to circumvent these problems, and obtain a semantic version of SQEMA for which we could prove d-persistence, we could try and find a syntactically correct equivalent of the downward-monotone sequents involved in the application of the rule, and substitute with it *before* we apply the rule. Accordingly, we will refer to the following rule as the *semantic Ackermann-rule with replacement*, and to the version of SQEMA obtained by replacing the ordinary Ackermann-rule with this rule as SSQEMA^r.

$$\text{The system} \quad \left\| \begin{array}{l} A_1 \Rightarrow p, \\ \vdots \\ A_n \Rightarrow p, \\ B_1(p), \\ \vdots \\ B_m(p), \end{array} \right. \quad \text{is replaced by} \quad \| \; B_1'((A_1 \vee \ldots \vee A_n)/p)$$

where:

1. p does not occur in $A_1, \ldots, A_n$,
2. $\mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$ is downwards monotone in p, and
3. $B_1'(p)$ is a sequent such that
   (a) $\mathsf{Form}(B_1'(p)) \equiv_{\text{sem}} \mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$,
   (b) $\mathsf{Form}(B_1'(p))$ is negative in p, and
   (c) $\mathsf{Form}(B_1'(p))$ is syntactically open.

Once again, a few comments are in order:

1. As in the case of the rule without replacement, the conditions of applicability mix syntactic and semantic criteria.
2. While a suitable $B'(p)$ always exists, as will be proven subsequently, the rule does not specify any particular method for obtaining such.

It its clear that SSQEMA^r will preserve transformation equivalence (definition 2.4.2). Moreover, a simple induction proves the following analogue of lemma 2.5.21:

Lemma 6.1.4 *During the entire (successful or unsuccessful) execution of SSQEMA^r on any $\mathcal{L}$ input formula, all antecedents of all non-diamond-link sequents are syntactically closed formulae, while all consequents of all non-diamond-link sequents are syntactically open.*

Hence, by a trivial modification of proposition 2.5.22, SSQEMA^r also preserves ad-transformation equivalence (recall definition 2.5.7). We now have the following theorem:

Theorem 6.1.5 *All $\mathcal{L}$-formulae on which SSQEMA^r succeeds are locally first-order definable and locally d-persistent.*

Here are two examples illustrating the some aspects of the execution of SSQEMA^r.

Example 6.1.6 Consider the input formula $\neg(\Box(\Box\Diamond(p \vee q) \vee \Diamond\Diamond\neg p \vee q) \wedge \Box p \wedge \Box\Diamond\neg q)$. The corresponding initial system, after application of the $\wedge$ and $\Box$-rules, is

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box(\Diamond p \vee \Diamond q) \vee \Diamond\Diamond\neg p \vee q) \\ \Diamond^{-1}\mathbf{i} \Rightarrow p \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

The system cannot be solved for q, so we try to eliminate p. The first sequent is downwards monotone in p, since $\Box(\Box(\Diamond p \vee \Diamond q) \vee \Diamond\Diamond\neg p \vee q) \equiv_{\mathrm{sem}} \Box(\Box\Diamond\top \vee \Diamond\Diamond\neg p \vee q)$. Accordingly, we replace $\mathbf{i} \Rightarrow \Box(\Box(\Diamond p \vee \Diamond q) \vee \Diamond\Diamond\neg p \vee q)$ with $\mathbf{i} \Rightarrow \Box(\Box\Diamond\top \vee \Diamond\Diamond\neg p \vee q)$, and apply the Ackermann-rule to obtain the system:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box\Diamond\top \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i} \vee q) \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

Next, solving for q we obtain

$$\left\| \begin{array}{l} \Diamond^{-1}\mathbf{i} \wedge \neg(\Box\Diamond\top \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i}) \Rightarrow q \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

A final application of the Ackermann-rule eliminates p. A few remarks are in order. Firstly, in each system obtained, each sequent has a syntactically closed antecedent and a syntactically closed consequent. Secondly, it should be clear that SQEMA will fail on this formula. Thirdly, suppose we had tried to reduce the input formula without replacement, i.e. with SSQEMA. Then, after the first application of the Ackermann-rule we would have obtained the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box(\Box(\Diamond\Diamond^{-1}\mathbf{i} \vee \Diamond q) \vee \Diamond\Diamond\Box^{-1}\neg\mathbf{i} \vee q) \\ \mathbf{i} \Rightarrow \Box\Diamond\neg q \end{array} \right. .$$

This system cannot be solved for q. However, note that the initial system is in fact downwards monotone in q. Hence, if we were to strengthen step Transform.1 to replace all variables in which the system is upwards (respectively, downwards) with $\top$ (respectively, $\bot$) then SSQEMA would succeed. $\lhd$

Example 6.1.7 Consider the input formula $\neg(\Box((\neg q \vee \neg p \vee \Diamond p) \wedge \Diamond\neg r) \wedge \Box(\neg p \vee \Box r) \wedge \Box q \wedge p)$. Once again, SQEMA will fail on this input. Let us see if SSQEMA^r fares any better. After a few applications of the $\wedge$-rule the initial system is transformed into

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Box((\neg q \vee \neg p \vee \Diamond p) \wedge \Diamond\neg r) \\ \mathbf{i} \Rightarrow \Box(\neg p \vee \Box r) \\ \mathbf{i} \Rightarrow \Box q \\ \mathbf{i} \Rightarrow p \end{array} \right. .$$

(Strictly speaking, conjunction should be distributed over disjunction on the first sequent, but as this makes no difference to the rest of the execution, we keep the sequent as it is for the sake of compactness of notation.) As the system stands, p cannot be eliminated, but q and r can. Indeed, solving the system for q and r yields

$$\left\|\begin{array}{l} \mathbf{i} \Rightarrow \Box((\neg q \vee \neg p \vee \Diamond p) \wedge \Diamond \neg r) \\ \Diamond^{-1}(\Diamond^{-1}\mathbf{i} \wedge \neg\neg p) \Rightarrow r \\ \Diamond^{-1}\mathbf{i} \Rightarrow q \\ \mathbf{i} \Rightarrow p \end{array}\right. ,$$

which, after two applications of the Ackermann-rule, becomes

$$\left\|\begin{array}{l} \mathbf{i} \Rightarrow \Box((\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)) \\ \mathbf{i} \Rightarrow p \end{array}\right. .$$

This is where SQEMA would get stuck. However, the sequent $\mathbf{i} \Rightarrow \Box((\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p))$ is downward monotone in p, as the formula $(\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$ is semantically equivalent to $\Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$. Applying the Ackermann-rule with replacement yields

$$\left\|\ \mathbf{i} \Rightarrow \Box(\Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg\mathbf{i}))\right. .$$

Note that, unlike the previous examples, the monotonicity of the formula $(\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$ essentially involves the fact that $\Diamond$ and $\Diamond^{-1}$ are inverses. $\lhd$

So the success of SSQEMA^r hinges on the availability of some mechanism to obtain suitable, syntactically open equivalents of given formulae which are negative in specified propositional variables. In the next section we will prove a Lyndon-type theorem that guarantees the existence of suitable such equivalents. In sections 6.3 and 6.4 we will develop algorithmic methods for obtaining the desired equivalents, based on (bi)simulation quantifiers.

6.2 On the Existence of Syntactically Correct Equivalents

In this section we prove versions of Lyndon's monotonicity theorem for syntactically open and syntactically closed $\mathcal{L}_r^n$-formulae. These theorems guarantee the existence of the equivalents demanded for substitution by the semantic Ackermann-rule with replacement. How these equivalents may be constructed is the topic of the next two sections. For a version of Lyndon's monotonicity theorem for the basic modal language, see [dRK97].

Definition 6.2.1 An $\mathcal{L}_r^n$*-bisimulation* is an ordinary tense bisimulation (the obvious adaptation of definition 1.5.6), but with the added requirement that points linked by it should agree not only on all propositional variables, but also on all nominals. We will use the notation $(\mathcal{M}, m) \rightleftarrows_{n,r} (\mathcal{N}, n)$ to indicate that there is an $\mathcal{L}_r^n$-bisimulation between models $\mathcal{M}$ and $\mathcal{N}$ linking points m and n. We will write $Z : (\mathcal{M}, m) \rightleftarrows_{n,r} (\mathcal{N}, n)$ if a particular $\mathcal{L}_r^n$-bisimulation Z is of importance.

The notion of an $\mathcal{L}_r^n$*-k-bisimulation* is the obvious relativization of an $\mathcal{L}_r^n$-bisimulation to depth k, analogous to definition 1.5.6. Accordingly we will write $(\mathcal{M}, m) \rightleftarrows_{n,r}^k (\mathcal{N}, n)$ to indicate that there is a $\mathcal{L}_r^n$-k-bisimulation between models $\mathcal{M}$ and $\mathcal{N}$ linking points m and n.

Note that we do not require the usual condition that all points which are the denotations of nominals to be linked by the $\mathcal{L}_r^n$-bisimulation to their counterparts in the other model, as this is not necessary for the local preservation of truth of $\mathcal{L}_r^n$-formulae. The following bisimulation notion is needed for the preservation of syntactically closed formulae, which are upwards monotone in certain variables. The following lemma is standard.

Lemma 6.2.2 *Suppose $\varphi \in \mathcal{L}_r^n$ with depth$(\varphi) \leq k$. Then, $(\mathcal{M}, m) \Vdash \varphi$ iff $(\mathcal{N}, n) \Vdash \varphi$, whenever $(\mathcal{M}, m) \rightleftarrows_{n,r}^k (\mathcal{N}, n)$.*

The following bisimulation notion is designed to preserve syntactically closed $\mathcal{L}_r^n$-formulae which are positive (or upward monotone) in certain propositional variables.

Definition 6.2.3 Let Θ be a possibly empty set of propositional variables and $k \in \omega$. A *syntactically closed* Θ-k-*simulation* relating a pointed model $(\mathcal{M}, m)$ to a pointed model $(\mathcal{N}, n)$ is any sequence of relations $Z_k \subseteq \cdots \subseteq Z_0 \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$, between the domains of the models, satisfying the following conditions:

(Link) mZ_kn.

(Asymmetric local harmony for Θ) If uZ_0v and $p \in \Theta$ then $(\mathcal{M}, u) \Vdash p$ implies $(\mathcal{N}, v) \Vdash p$.

(Asymmetric local harmony for nominals) if uZ_0v and $\mathbf{i} \in \mathsf{NOM}$ then $(\mathcal{M}, u) \Vdash \mathbf{i}$ implies $(\mathcal{N}, v) \Vdash \mathbf{i}$.

(Local harmony for propositional variables) if uZ_0v and $p \in \mathsf{PROP} - \Theta$, then $(\mathcal{M}, u) \Vdash p$ iff $(\mathcal{N}, v) \Vdash p$.

(Reversive Forth) if $uZ_{i+1}v$ and $R^{\mathcal{M}}uu'$, then $R^{\mathcal{N}}vv'$ for some $v' \in \mathcal{N}$ such that $u'Z_iv'$; similarly if $uZ_{i+1}v$ and $R^{\mathcal{M}}u'u$, then $R^{\mathcal{N}}v'v$ for some $v' \in \mathcal{N}$ such that $u'Z_iv'$.

(Non-Reversive Back) if $uZ_{i+1}v$ and $R^{\mathcal{N}}vv'$, then $R^{\mathcal{M}}uu'$ for some $u' \in \mathcal{M}$ such that $u'Z_iv'$.

We will use the notation $(\mathcal{M}, m) \rightrightarrows_{SC(\Theta)}^k (\mathcal{N}, n)$ to indicate that there exists a syntactically closed Θ-k-simulation relating $(\mathcal{M}, m)$ to $(\mathcal{N}, n)$. We will write $Z : (\mathcal{M}, m) \rightrightarrows_{SC(\Theta)}^k (\mathcal{N}, n)$ if a particular Θ-k-simulation Z is of importance.

Lemma 6.2.4 *Let Θ be a finite, possibly empty set of propositional variables. Any syntactically closed $\mathcal{L}_r^n$-formula φ of modal depth $\leq k$, which is positive in the propositional variables from Θ, is preserved under syntactically closed Θ-k-simulations.*

Proof. By structural induction on φ, written in negation normal form. QED

The next lemma strengthens lemma 6.2.4, by replacing *positivity* with *upward monotonicity*.

Lemma 6.2.5 *Let Θ be a finite, possibly empty set of propositional variables. Any syntactically closed $\mathcal{L}_r^n$-formula φ of modal depth $\leq k$, which is upwards monotone in the propositional variables from Θ, is preserved under syntactically closed Θ-k-simulations.*

Proof. Let φ satisfy the conditions of the lemma and let $Z_k \subseteq \cdots \subseteq Z_0 \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$ be a syntactically closed Θ-k-simulation between the models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$. Suppose that $(\mathcal{M}, m) \Vdash \varphi$.

Let the model $\mathcal{M} \ltimes \mathcal{N} = \langle W^{\ltimes}, R^{\ltimes}, V^{\ltimes} \rangle$ be defined as follows: $W^{\ltimes} = Z_0$; $R^{\ltimes}(u, v)(u', v')$ iff $R^{\mathcal{M}}uu'$ and $R^{\mathcal{N}}vv'$; $V^{\ltimes}(p) = \{(u, v) \in Z_0 \mid u \in V^{\mathcal{M}}(p)\}$ for all propositional variables p; and $V^{\ltimes}(\mathbf{j}) = \{(u, v) \in Z_0 \mid u \in V^{\mathcal{M}}(\mathbf{j})\}$ for all nominals $\mathbf{j}$. Note that for every nominal $\mathbf{j}$ whose denotation in $\mathcal{M}$ is linked to a point in $\mathcal{N}$ by Z_0, $V^{\ltimes}(\mathbf{j})$ is a singleton due to the asymmetric local harmony for nominals. All other nominals, however, are interpreted by $V^{\ltimes}$ as $\emptyset$; to remedy this defect we tacitly add to $\mathcal{M} \ltimes \mathcal{N}$ a new point, unrelated to any other by the accessibility relation, where we interpret all those nominals, as well as all propositional variables. The following hold:

(i) $(m, n) \in W^{\ltimes}$, by construction.

(ii) $(\mathcal{M}, m) \rightleftarrows_{r,n}^{k} (\mathcal{M} \ltimes \mathcal{N}, (m, n))$, by routine verification that $Z'_k \subseteq \cdots \subseteq Z'_0$ with $Z'_i = \{(u, (u, v)) \mid (u, v) \in Z_i\}$ satisfies definition 6.2.1.

(iii) Hence, $(\mathcal{M} \ltimes \mathcal{N}, (m, n)) \Vdash \varphi$, by lemma 6.2.2.

(iv) Moreover, $(\mathcal{M} \ltimes \mathcal{N}, (m, n)) \rightrightarrows_{SC(\Theta)}^{k} (\mathcal{N}, n)$, by routine verification that $Z''_k \subseteq \cdots \subseteq Z''_0$ with $Z''_i = \{((u, v), v) \mid (u, v) \in Z_i\}$ satisfies definition 6.2.3.

Let $\mathcal{M} \curlywedge \mathcal{N}$ be obtained from $\mathcal{M} \ltimes \mathcal{N}$ by extending the valuations of the propositional variables in $p \in \Theta$ as follows: for every point in $(u, v) \in \mathcal{M} \curlywedge \mathcal{N}$, let $(u, v) \in V^{\curlywedge}(p)$ iff $v \in V^{\mathcal{N}}(p)$. Note that $V^{\ltimes}(p) \subseteq V^{\curlywedge}(p)$. It follows from the upward monotonicity of φ in the variables from Θ that $(\mathcal{M} \curlywedge \mathcal{N}, (m, n)) \Vdash \varphi$. Finally, it is straightforward to check that $(\mathcal{M} \curlywedge \mathcal{N}, (m, n)) \rightrightarrows_{SC(\emptyset)}^{k} (\mathcal{N}, n)$. Hence, by lemma 6.2.5, $(\mathcal{N}, n) \Vdash \varphi$. QED

Let us denote by $(\mathcal{L}_r^n)_{SC(\Theta)}^k$ the set of all syntactically closed $\mathcal{L}_r^n$-formulae, positive in all propositional variables is Θ and of modal depth at most k. We will write $(\mathcal{M}, m) \Rrightarrow_{SC(\Theta)}^{k} (\mathcal{N}, n)$ if $(\mathcal{M}, m) \Vdash \varphi$ implies $(\mathcal{N}, n) \Vdash \varphi$ for all $\varphi \in (\mathcal{L}_r^n)_{SC(\Theta)}^k$. Note, that $\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^k$ iff $\Diamond\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^{k+1}$ iff $\Box\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^{k+1}$ iff $\Diamond^{-1}\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^{k+1}$.

Remark 6.2.6 $(\mathcal{M}, m) \Rrightarrow_{SC(\Theta)}^{k} (\mathcal{N}, n)$ iff $(\mathcal{N}, n) \Vdash \psi$ implies $(\mathcal{M}, m) \Vdash \psi$ for all ψ such that $\neg\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^k$, i.e., iff all syntactically open formulae, negative in the propositional variables in Θ and of modal depth at most k are preserved in passing from $(\mathcal{N}, n)$ to $(\mathcal{M}, m)$.

In what follows we will have to be more precise about the propositional variables and nominals that occur in the language. We will therefore denote by $\mathcal{L}_r^n(\Phi, \Psi)$ the language $\mathcal{L}_r^n$ built over the propositional variables in Φ and the nominals in Ψ. $(\mathcal{L}_r^n(\Phi, \Psi))_{SC(\Theta)}^k$ is accordingly the restriction of $\mathcal{L}_r^n(\Phi, \Psi)$ to $(\mathcal{L}_r^n)_{SC(\Theta)}^k$. The relations $\rightrightarrows_{SC(\Theta)(\Phi,\Psi)}^{k}$ and $\Rrightarrow_{SC(\Theta)(\Phi,\Psi)}^{k}$ are similarly generalized from $\rightrightarrows_{SC(\Theta)}^{k}$ and $\Rrightarrow_{SC(\Theta)}^{k}$.

Lemma 6.2.7 *For any pointed models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$, set of propositional variables Θ, finite sets Φ and Ψ respectively of propositional variables and nominals, and $k \in \omega$,*

$$(\mathcal{M}, m) \rightrightarrows^{k}_{SC(\Theta)(\Phi,\Psi)} (\mathcal{N}, n) \quad \textit{iff} \quad (\mathcal{M}, m) \Rrightarrow^{k}_{SC(\Theta)(\Phi,\Psi)} (\mathcal{N}, n).$$

Proof. The left-to-right direction is lemma 6.2.4. In the rest of the proof we suppress reference to Φ and Ψ — the only important fact about them is that they are finite, and hence that $(\mathcal{L}^n_r)^k_{SC(\Theta)}(\Phi, \Psi)$ is finite, modulo equivalence. We prove the right-to-left direction. Suppose that $(\mathcal{M}, m) \Rrightarrow^{k}_{SC(\Theta)} (\mathcal{N}, n)$ and let

$$Z_i = \{(u, v) \in W^{\mathcal{M}} \times W^{\mathcal{N}} \mid (\mathcal{M}, u) \Rrightarrow^{i}_{SC(\Theta)} (\mathcal{N}, v)\},$$

for all $0 \leq i \leq k$. We will show that, $Z_k \subseteq \cdots \subseteq Z_0$ is a Θ-k-simulation linking m and n. By construction, $(m, n) \in Z_k$. It should also be clear that the symmetric and asymmetric local harmony clauses are satisfied by any $(u, v) \in Z_0$. Suppose that $(u, v) \in Z_i$, for some $0 < i \leq k$. We must show that (u, v) satisfies the back and forth-clauses required by the definition.

Suppose that $R^{\mathcal{M}}uu'$. Let $SC(\Theta)^{i-1}(u') = \{\psi \in (\mathcal{L}^n_r)^{i-1}_{SC(\Theta)} \mid (\mathcal{M}, u') \Vdash \psi\}$, i.e. the set of all $(\mathcal{L}^n_r)^{i-1}_{SC(\Theta)}$-formulae true at u'. We may assume that $SC(\Theta)^{i-1}(u')$ is finite. Then $\Diamond \bigwedge SC(\Theta)^{i-1}(u')$ is an $(\mathcal{L}^n_r)^{i}_{SC(\Theta)}$-formula, such that $(\mathcal{M}, u) \Vdash \Diamond \bigwedge SC(\Theta)^{i-1}(u')$. Hence, $(\mathcal{N}, v) \Vdash \Diamond \bigwedge SC(\Theta)^{i-1}(u')$, that is to say, v has a $R^{\mathcal{N}}$-successor, say v', such that $(\mathcal{N}, v') \Vdash \bigwedge SC(\Theta)^{i-1}(u')$. It follows that $(\mathcal{M}, u') \Rrightarrow^{i-1}_{SC(\Theta)} (\mathcal{N}, v')$, and hence that $(u', v') \in Z_{i-1}$. This proves half of the reversive forth-clause. The other half is symmetric, using the formula $\Diamond^{-1} \bigwedge SC(\Theta)^{i-1}(u')$ for u' an $R^{\mathcal{M}}$-predecessor of u.

Now for the non-reversive back-clause. Suppose that $(u, v) \in Z_i$ and that $R^{\mathcal{N}}vv'$. Let $SO(\Theta)^{i-1}(v') = \{\psi : \neg\psi \in (\mathcal{L}^n_r)^{i-1}_{SC(\Theta)} \ \& \ (\mathcal{N}, v') \Vdash \psi\}$, i.e. the set of all syntactically open formulae of modal depth at most $(i-1)$ and negative in the propositional variables in Θ which are true at v'. Again, we may assume that $SO(\Theta)^{i-1}(v')$ is finite. Then $\Diamond \bigwedge SO(\Theta)^{i-1}(v')$ is a syntactically open formula of modal depth at most i, negative in the propositional variables from Θ, and hence, by remark 6.2.6, $(\mathcal{M}, u) \Vdash \Diamond \bigwedge SO(\Theta)^{i-1}(v')$. Hence, there is a u' such that $R^{\mathcal{M}}uu'$ and $(\mathcal{M}, u') \Vdash \bigwedge SO(\Theta)^{i-1}(v')$. Again, by remark 6.2.6 it follows that $(\mathcal{M}, u') \Rrightarrow^{i-1}_{SC(\Theta)} (\mathcal{N}, v')$ and hence that $(u', v') \in Z_{i-1}$. Note, by the way, that we would not be able to prove a reversive back-clause in a similar way, since $\Diamond^{-1} \bigwedge SO(\Theta)^{i-1}(v')$ is not syntactically open. QED

Theorem 6.2.8 (Lyndon's monotonicity theorem for syntactically closed formulae) *A syntactically closed formula $\varphi \in \mathcal{L}^n_r$ is upward monotone in the propositional variables in a set Θ if and only if it is semantically equivalent to a syntactically closed formula $\varphi' \in \mathcal{L}^n_r$ which is positive in the propositional variables in Θ and such that $\mathsf{PROP}(\varphi') \subseteq \mathsf{PROP}(\varphi)$, $\mathsf{NOM}(\varphi') \subseteq \mathsf{NOM}(\varphi)$ and $depth(\varphi') \leq depth(\varphi)$.*

Proof. The right-to-left direction of the bi-implication is immediate. So, assume that $\varphi \in \mathcal{L}^n_r$ is syntactically closed and upwards monotone in the propositional variables Θ, and suppose that $depth(\varphi) = k$. Let

$$\mathrm{CONS}(\varphi) = \{\psi \in (\mathcal{L}^n_r(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))^k_{SC(\Theta)} \mid \ \Vdash \varphi \to \psi\}.$$

Claim: CONS(φ) is a finite set, modulo semantic equivalence.

Proof of Claim: It is sufficient to show that $\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))^k$ is a finite set. We proceed by induction on k. For $k = 0$ we are dealing with the set of all boolean combinations of elements of $\mathsf{PROP}(\varphi) \cup \mathsf{NOM}(\varphi)$. But since there are only a finite number of truth functions on any given finite number of arguments, there are, modulo semantic equivalence, only a finite number of boolean combinations of the members of any finite set of formulae.

Suppose the claim holds for $k = n$. By appealing to the distributive identities any formula in $\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))^{n+1}$ can be equivalently written as a disjunction of formulae of the from

$$A \wedge \bigwedge \Diamond B_i \wedge \bigwedge \Box C_i \wedge \bigwedge \Diamond^{-1} D_i \wedge \bigwedge \Box^{-1} E_i$$

where A as well as all B_i, C_i, D_i and E_i are in $\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))^n$. But by the inductive hypothesis the latter set is finite, modulo semantic equivalence. The claim then follows since, as remarked above, modulo semantic equivalence there are only finitely many different boolean combinations of any finite number of formulae. ◀

The proof is complete once we can show that CONS(φ) $\Vdash \varphi$, since we can then take φ' to be $\bigwedge$CONS(φ). To this end, suppose that $(\mathcal{N}, n) \Vdash$ CONS(φ). Let $N = \{\psi \mid \neg\psi \in (\mathcal{L}_r^n)^k_{SC(\Theta)}$ & $(\mathcal{N}, n) \Vdash \psi\}$. Then $N \cup \{\varphi\}$ is satisfiable, for otherwise $N \Vdash^{loc} \neg\varphi$, i.e $\bigwedge N \Vdash^{loc} \neg\varphi$. But then $\varphi \Vdash \bigvee\{\neg\psi \mid \psi \in N\}$ and $\bigvee\{\neg\psi \mid \psi \in N\} \in$ CONS(φ) — a contradiction.

Let $(\mathcal{M}, m) \Vdash N \cup \{\varphi\}$. Then, by remark 6.2.6, $(\mathcal{M}, m) \Rrightarrow^k_{SC(\Theta)} (\mathcal{N}, n)$, hence, by lemma 6.2.7 we have $(\mathcal{M}, m) \rightrightarrows^k_{SC(\Theta)} (\mathcal{N}, n)$, and then by lemma 6.2.5 it follows that $(\mathcal{N}, n) \Vdash \varphi$. QED

By taking negations, we obtain the following corollary.

Theorem 6.2.9 (Lyndon's monotonicity theorem for syntactically open formulae) *A syntactically open formula $\varphi \in \mathcal{L}_r^n$ is downward monotone in the propositional variables in a set Θ if and only if it is semantically equivalent to a syntactically open formula $\varphi' \in \mathcal{L}_r^n$ which is negative in the propositional variables in Θ and such that $\mathsf{PROP}(\varphi') \subseteq \mathsf{PROP}(\varphi)$, $\mathsf{NOM}(\varphi') \subseteq \mathsf{NOM}(\varphi)$ and depth$(\varphi') \leq$ depth(φ).*

6.3 Negative equivalents for separately monotone formulae

In many of the examples above, the monotonicity of formulae involved in the application of the Ackermann-rule did not depend on the proper interpretation of the inverse modalities $\Diamond^{-1}$ and $\Box^{-1}$ as inverses of $\Diamond$ and $\Box$. For example, the downwards monotonicity of $\Box^{-1}\neg\mathbf{i} \vee (\Diamond\Box p \wedge \Box\Box\neg p)$ in p can be detected by looking at $\Box_2\neg\mathbf{i} \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$. Moreover, the fact that $\mathbf{i}$ is a nominal is also irrelevant — $\Box_2\neg r \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$ is downward monotone in p for any propositional variable r.

With this observation in mind, we introduce the following terminology and definitions. We will refer to the bimodal language with two diamonds $\Diamond_1$ and $\Diamond_2$ as $\mathcal{L}_2$.

Definition 6.3.1 Given a formula $\varphi \in \mathcal{L}_r^n$ the *separation* of φ, denoted $\mathsf{Sep}(\varphi)$, is the $\mathcal{L}_2$-formula obtained by

1. replacing every occurrence of $\Diamond$ and $\Box$ in φ with $\Diamond_1$ and $\Box_1$, respectively,
2. replacing every occurrence of $\Diamond^{-1}$ and $\Box^{-1}$ in φ with $\Diamond_2$ and $\Box_2$, respectively, and
3. uniformly substituting a fresh propositional variable for every nominal occurring in φ.

For example, $\mathsf{Sep}(\Box^{-1}\neg\mathbf{i} \vee (\Diamond\Box p \wedge \Box\Box\neg p))$ is $\Box_2\neg r \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$.

Definition 6.3.2 A $\mathcal{L}_r^n$-formula φ is *separately upward monotone* in a proposition variable p if $\mathsf{Sep}(\varphi)$ is upwards monotone in p. The notion of *separate downward monotonicity* is defined similarly.

Clearly separate monotonicity implies ordinary monotonicity. Recall that monotonicity and validity (and hence, satisfiability) problems for formulae are interreducible. Now, the validity problem for $\mathcal{L}_r^n$-formulae is EXPTIME-complete ([ABM00]), while those for $\mathcal{L}_r$ ([Spa93]) and for $\mathcal{L}_2$ ([HM92]) are PSPACE-complete. Hence, with the aim of minimizing computational cost, it might be wise to test formula for separate monotonicity first, and only if that fails to test for ordinary monotonicity. In section 6.4 we introduce an intermediate monotonicity notion which requires the testing of $\mathcal{L}_r$-formulae for validity.

In this section we present a method for finding negative (positive) syntactically open (closed) equivalents for separately downwards (upwards) monotone formulae. The method will be based on an adaptation of the method of *bisimulation quantifiers*. The idea originates from the '*Pitss quantifiers*' of [Pit92]. Bisimulation quantifiers have been used to prove uniform interpolation results for the modal μ-calculus in [DL02] and for some modal logics in [Vis96] and [Ghi95]. The normal form used is inspired by that in [tC05a] and related to that introduced in [JW95].

6.3.1 Disjunctive forms

If S is a finite, possibly empty, set of formulae, define ∇S as shorthand for

$$\bigwedge_{\varphi \in S} \Diamond_1 \varphi \wedge \Box_1 \bigvee_{\varphi \in S} \varphi,$$

and $\triangle S$ as shorthand for

$$\bigwedge_{\varphi \in S} \Diamond_2 \varphi.$$

Note the asymmetry between these definitions — ∇S and $\triangle S$ are defined like this because they will be used to write the separations of syntactically closed formulae. In the case of singleton sets S or S', we will often write $\nabla\varphi$ and $\triangle\varphi$ for $\nabla\{\varphi\}$ and $\triangle\{\varphi\}$, respectively.

Propositional variables and their negations will be called *literals*. Given a propositional variable p, p and $\neg p$ are *p-literals*. Moreover, p and $\neg p$ are *complementary literals*. For a set, Θ, of propositional variables, a Θ-literal is any p-literal for some $p \in \Theta$.

Definition 6.3.3 The $\mathcal{L}_2$-formulae in *disjunctive form* are given recursively by

$$\varphi ::= \bot \mid \top \mid \chi \wedge \nabla S \wedge \triangle S' \mid \varphi \vee \psi,$$

where χ is a (possibly empty) conjunction of literals, S a (possibly empty) and S' a non-empty set of formulae in disjunctive form. As usual, we identify the empty conjunction with $\top$, and the empty disjunction with $\bot$. Note that the forms $\chi \wedge \nabla S$, $\nabla S \wedge \triangle S'$ and ∇S can be seen as special cases of $\chi \wedge \nabla S \wedge \triangle S'$ with respectively S', χ, or both, empty.

We will call a $\mathcal{L}_2$-formula *syntactically closed* if it contains no positive occurrence of $\Box_2$. (Since we will always be careful to specify in which language we work, this reuse of terminology should cause no confusion. Moreover, in terms of definition 2.5.16, all $\mathcal{L}_2$-formulae are syntactically closed, rendering that notion meaningless for such formulae.) Clearly the separation of any syntactically closed $\mathcal{L}_r^n$-formula will be a syntactically closed $\mathcal{L}_2$-formula. Next we define a translation, $(\cdot)^\star$ of syntactically closed $\mathcal{L}_2$-formulae, written in negation normal form, into disjunctive form. When reading this definition, it is useful to bear the following equivalences in mind: $\nabla\emptyset \equiv_{\text{sem}} \Box_1\bot$, $\nabla\{\varphi, \top\} \equiv_{\text{sem}} \Diamond_1\varphi \wedge \Diamond_1\top \wedge \Box_1(\varphi \vee \top) \equiv_{\text{sem}} \Diamond_1\varphi$, $\nabla\{\top\} \equiv \Diamond_1\top$ and $\varphi \equiv_{\text{sem}} ((\varphi \wedge \Diamond_1\top) \vee (\varphi \wedge \Box_1\bot)$.

$$\begin{aligned}
\top^\star &= \top \\
\bot^\star &= \bot \\
lit^\star &= (lit \wedge \nabla\emptyset) \vee (lit \wedge \nabla\top) \quad \text{for any literal } lit \\
(\varphi \vee \psi)^\star &= \varphi^\star \vee \psi^\star \\
(\Diamond_1\varphi)^\star &= \nabla\{\varphi^\star, \top\} \\
(\Diamond_2\varphi)^\star &= (\nabla\emptyset \wedge \triangle\varphi^\star) \vee (\nabla\top \wedge \triangle\varphi^\star) \\
(\Box_1\varphi)^\star &= \nabla\emptyset \vee \nabla\varphi^\star
\end{aligned}$$

The case for conjunction is more complicated. Consider a formula of the form $\bigwedge S$. If S is such that $S = S' \cup \{\top\}$, $S = S' \cup \{\bot\}$, or $S = S' \cup \{\varphi \vee \psi\}$ we translate as follows

$$\begin{aligned}
(\bigwedge(S' \cup \{\top\}))^\star &= (\bigwedge S')^\star \\
(\bigwedge(S' \cup \{\bot\}))^\star &= \bot \\
(\bigwedge(S' \cup \{\varphi \vee \psi\}))^\star &= (\bigwedge(S' \cup \{\varphi\}))^\star \vee (\bigwedge(S' \cup \{\psi\}))^\star
\end{aligned}$$

Note that in the last case above we are in effect distributing the conjunction over the disjunction. If S does not contain $\top$, $\bot$ or a disjunction, it means that every formula in S is either a literal or a formula of the form $\Diamond_1\psi$, $\Box_1\psi$, or $\Diamond_2\psi$. We form the following sets:

$$\begin{aligned}
S_{\Diamond_1} &= \{\psi \mid \Diamond_1\psi \in S\} \\
S_{\Box_1} &= \{\psi \mid \Box_1\psi \in S\} \\
S_{\Diamond_2} &= \{\psi \mid \Diamond_2\psi \in S\}
\end{aligned}$$

Lastly, let S_{lit} be the subset of all literals in S. If $S_{\Diamond_1} \neq \emptyset$, then the intuition is that any point satisfying $\bigwedge S$ must satisfy each member of S_{lit}, every member of $S_{\Diamond_2}$ must be satisfied at some R_2-successor, and every member of $S_{\Diamond_1}$ must be satisfied at some R_1-successor which also satisfies all members of $S_{\Box_1}$. We must also take into account the fact that there may be R_1-successors not satisfying any member of $S_{\Diamond_1}$, but which still have to satisfy all members of $S_{\Box_1}$. Hence, if $S_{\Diamond_1} \neq \emptyset$, we translate thus:

$$(\bigwedge S)^\star = \bigwedge S_{lit} \wedge \nabla\{(\varphi \wedge \bigwedge S_{\Box_1})^\star \mid \varphi \in S_{\Diamond_1} \cup \{\top\}\} \wedge \triangle\{\psi^\star \mid \psi \in S_{\Diamond_2}\}.$$

If, on the other hand, $S_{\Diamond_1} = \emptyset$, points satisfying the formula can either have no R_1-successors, or have R_1-successors, each satisfying every member of $S_{\Box_1}$. Hence, if $S_{\Diamond_1} = \emptyset$, let

$$(\bigwedge S)^\star = (\bigwedge S_{lit} \wedge \nabla\emptyset \wedge \triangle\{\psi^\star \mid \psi \in S_{\Diamond_2}\}) \vee (\bigwedge S_{lit} \wedge \nabla\{(\bigwedge S_{\Box_1})^\star\} \wedge \triangle\{\psi^\star \mid \psi \in S_{\Diamond_2}\}).$$

It should be clear that $\varphi \equiv_{\text{sem}} \varphi^\star$ for every syntactically closed $\mathcal{L}_2$-formula φ in negation normal form. Here is an example:

Example 6.3.4 Consider the formula $r \wedge \Diamond_1(\Diamond_1\Box_1\neg p \wedge \Box_1\Box_1 p \wedge \neg q)$. Since it contains no occurrences of $\Diamond_2$, we will omit the subscripts and simply write $\Diamond$ and $\Box$ for $\Diamond_1$ and $\Box_1$, respectively. It is translated into disjunctive form as follows:

$$\begin{aligned}
&(r \wedge \Diamond(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q))^\star \\
=&r \wedge \nabla\{(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q)^\star, \top\} \\
=&r \wedge \nabla\{\neg q \wedge \nabla\{(\Box\neg p \wedge \Box p)^\star, (\Box p)^\star\}, \top\} \\
=&r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\}
\end{aligned}$$

$\lhd$

6.3.2 Simulation quantifiers and biased simulations

Via disjunctive forms and the following definition we will transform upward monotone syntactically closed formulae into positive ones.

Definition 6.3.5 Let φ be an $\mathcal{L}_2$-formula in disjunctive form and $\overline{p}$ a vector of propositional variables. We define $\exists^+\overline{p}.\varphi$ inductively as follows:

$$\begin{aligned}
\exists^+\overline{p}.\bot &= \bot \\
\exists^+\overline{p}.\top &= \top \\
\exists^+\overline{p}.(\chi \wedge \nabla S \wedge \triangle S') &= \chi' \wedge \nabla\{\exists^+\overline{p}.\psi \mid \psi \in S\} \wedge \triangle\{\exists^+\overline{p}.\psi \mid \psi \in S'\} \\
\exists^+\overline{p}.(\varphi \vee \psi) &= \exists^+\overline{p}.\varphi \vee \exists^+\overline{p}.\psi
\end{aligned}$$

where χ' is $\bot$ when χ is inconsistent (i.e. when χ contains complementary literals), or otherwise, if χ is consistent, χ' is obtained from χ by removing (by simply deleting) all occurrences of negative $\overline{p}$-literals. $\exists^+\overline{p}$ is called a *simulation quantifier*.

Note that $\exists^+\overline{p}.\varphi$ is positive in all variables in $\overline{p}$. We want to show that $\exists^+p.\varphi \equiv_{\text{sem}} \varphi$ for all formulas φ that are upward monotone in p. To that aim the following definition, which is essentially a separated version of a syntactically closed Θ-simulation (definition 6.2.3).

Definition 6.3.6 Let $\mathcal{M} = (W^{\mathcal{M}}, R_1^{\mathcal{M}}, R_2^{\mathcal{M}}, V^{\mathcal{M}})$ and $\mathcal{N} = (W^{\mathcal{N}}, R_1^{\mathcal{N}}, R_2^{\mathcal{N}}, V^{\mathcal{N}})$ be $\mathcal{L}_2$-models. Let Θ be sets of propositional variables. A Θ-*biased simulation* between $\mathcal{M}$ and $\mathcal{N}$ is a nonempty binary relation $Z \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$ satisfying, for all $(u, v) \in W^{\mathcal{M}} \times W^{\mathcal{N}}$ such that uZv, the following conditions:

(local harmony) $(\mathcal{M}, u) \Vdash p$ iff $(\mathcal{N}, v) \Vdash p$ for all propositional variables $p \notin \Theta$,

(asymmetric local harmony) $(\mathcal{M}, u) \Vdash p$ only if $(\mathcal{N}, v) \Vdash p$, for all propositional variables $p \in \Theta$,

(symmetric forth) if $R_1^{\mathcal{M}}uu'$ (respectively, $R_2^{\mathcal{M}}uu'$) then there exists a point $v' \in W^{\mathcal{N}}$ such that $u'Zv'$ and $R_1^{\mathcal{N}}vv'$ (respectively, $R_2^{\mathcal{N}}vv'$), and

(asymmetric back) if $R_1^{\mathcal{N}}vv'$ then there exists a point $u' \in W^{\mathcal{M}}$ such that $u'Zv'$ and $R_1^{\mathcal{N}}uu'$.

We will write $\mathcal{M} \hookrightarrow_\Theta \mathcal{N}$ if there exists a Θ-biased simulation between models $\mathcal{M}$ and $\mathcal{N}$, or $(\mathcal{M}, m) \hookrightarrow_\Theta (\mathcal{N}, n)$ if there is a Θ-biased simulation linking m and n.

A straightforward adaptation of the proof of lemma 6.2.5 establishes the next lemma.

Lemma 6.3.7 *For all $\mathcal{L}_2$-models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$ such that $(\mathcal{M}, m) \hookrightarrow_\Theta (\mathcal{N}, n)$, and all syntactically closed $\mathcal{L}_2$-formulae φ, which are upward monotone in the variables in Θ, it holds that $(\mathcal{M}, m) \Vdash \varphi$ only if $(\mathcal{N}, n) \Vdash \varphi$.*

Lemma 6.3.8 *Let $\varphi \in \mathcal{L}_2$ be a syntactically closed formula in disjunctive from and $\overline{p}$ a vector of propositional variables. Then, $(\mathcal{M}, m) \Vdash \varphi$ implies $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$.*

Proof. By induction on φ. QED

The next theorem motivates why we call $\exists^+\overline{p}$ a 'simulation quantifier':

Proposition 6.3.9 *Let $\varphi \in \mathcal{L}_2$ be a syntactically closed formula in disjunctive from. Then, for any model $(\mathcal{N}, n)$ and any vector of propositional variables $\overline{p}$,*

$$(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$$

if and only if there exists a model $(\mathcal{M}, m)$ such that

$$(\mathcal{M}, m) \Vdash \varphi \quad \textit{and} \quad (\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n).$$

Proof. We proceed by induction on φ. The base case for $\top$ is trivial, as is the inductive step for φ of the form $\psi_1 \vee \psi_2$. We consider the case for φ of the form $\chi \wedge \nabla S \wedge \triangle S'$.

The bottom-to-top direction is easy. By lemma 6.3.8, $(\mathcal{M}, m) \Vdash \varphi$ implies $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$. Also note that $\exists^+\overline{p}.\varphi$ is positive in all propositional variables in $\overline{p}$. We can now appeal to lemma 6.3.7, and conclude that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$.

Conversely, suppose that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$. By the inductive hypothesis it follows that, for each pair (ψ, s) such that $\psi \in S$, $R_1^{\mathcal{N}}ns$ and $(\mathcal{N}, s) \Vdash \exists^+\overline{p}.\psi$, there exists a pointed model $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)})$ such that $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \Vdash \psi$ and $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$. Moreover, since every $R_1^{\mathcal{N}}$-successor s of n satisfies $\exists^+\overline{p}.\psi$ for some formula $\psi \in S$, we have that $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$ with $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \Vdash \psi$ for some $\psi \in S$.

Also by the inductive hypothesis, for every $\psi \in S'$ there exists a point $s \in W^{\mathcal{N}}$ and a pointed model $(\mathcal{M}_\psi, m_\psi)$ such that $R_2^{\mathcal{N}} ns$, $(\mathcal{M}_\psi, m_\psi) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$ and $(\mathcal{M}_\psi, m_\psi) \Vdash \psi$.

Now we construct the desired model $(\mathcal{M}, m)$ by first taking the disjoint union of the models in the sets

$$\left\{(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \mid \psi \in S, R_1^{\mathcal{N}} ns, (\mathcal{N}, s) \Vdash \exists^+\overline{p}.\psi\right\}$$

and

$$\left\{(\mathcal{M}_\psi, m_\psi) \mid \psi \in S'\right\}.$$

To this disjoint union we add a new point m and make it an R_1-predecessor of each $m_{(\psi,s)}$, and an R_2-predecessor of each m_ψ. To complete the model we make all propositional variables occurring positively in χ true at m while all other propositional variables are declared false there. By construction $(\mathcal{M}, m) \Vdash \varphi$ and $(\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n)$. QED

Theorem 6.3.10 *Let $\varphi \in \mathcal{L}_2$ be a syntactically closed formula in disjunctive from which is upward monotone in $\overline{p}$. Then $\varphi \equiv_{\mathrm{sem}} \exists^+\overline{p}.\varphi$.*

Proof. As remarked before, $\Vdash \varphi \to \exists^+\overline{p}.\varphi$. Conversely, suppose that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$. By proposition 6.3.9 there exists a model $(\mathcal{M}, m)$ such that $(\mathcal{M}, m) \Vdash \varphi$ and $(\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n)$. But, by lemma 6.3.7, φ is preserved under $\overline{p}$-biased simulations, i.e. $(\mathcal{N}, n) \Vdash \varphi$. QED

Theorem 6.3.10 gives us a procedure to compute positive equivalents for upward monotone syntactically closed $\mathcal{L}_2$-formulae, written in disjunctive form. This is easily converted into a procedure for computing negative equivalents for separately downward monotone syntactically open $\mathcal{L}_r^n$-formulae. To be precise, suppose that $\varphi \in \mathcal{L}_r^n$ is syntactically open and separately downward monotone in the propositional variable p. We compute the desired equivalent of φ as follows:

1. **negation:** Negate φ and apply the usual procedure to rewrite the $\neg\varphi$ in negation normal form, obtaining φ'. The formula φ' is syntactically closed and separately upward monotone in p.

2. **separation:** Separate φ' by calculation $\mathsf{Sep}(\varphi')$. The formula $\mathsf{Sep}(\varphi')$ will be a syntactically closed $\mathcal{L}_2$-formula which is upward monotone in p.

3. **disjunctive form:** Transform $\mathsf{Sep}(\varphi')$ into disjunctive form by applying the translation $(\cdot)^\star$, i.e. by calculating $(\mathsf{Sep}(\varphi'))^\star$.

4. **eliminate negative p-occurrences:** Calculate $\exists^+p.(\mathsf{Sep}(\varphi'))^\star$. This formula is positive in p.

5. **obtain positive $\mathcal{L}_r^n$-equivalent:** Reverse step 3 as far as possible by applying the inverse of the translation function $(\cdot)^\star$ and the definitions of ∇S and $\triangle S'$. Lastly obtain an $\mathcal{L}_r^n$-formula by applying the inverse of Sep.

6. second negation: Negate the resulting formula again to obtain a syntactically open formula, negative in p, and semantically equivalent to φ.

We illustrate this procedure with an example.

Example 6.3.11 In example 6.1.1 we used the fact that the sequent $\mathbf{i} \Rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q)$, that is to say the formula $\gamma = \neg\mathbf{i} \vee \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q)$, was downward monotone in p. Indeed, it is even separately downward monotone in p, as $\mathsf{Sep}(\gamma) = \neg r \vee \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q)$ is downward monotone in p. (Since there are no inverse modalities involved in this formula, we can omit the subscripts in the separated from without risk of confusion.) Let us compute a negative equivalent for this formula using the method of simulation quantifiers, described above. Negating and rewriting in negation normal form we obtain $r \wedge \Diamond(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q)$. In example 6.3.4 this formula was translated into disjunctive form, thus:

$$\begin{aligned} &(r \wedge \Diamond(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q))^\star \\ =&r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \end{aligned}$$

Next, application of the simulation quantifier $\exists^+ p$ yields

$$\begin{aligned} &\exists^+ p.(r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\}) \\ =&r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\bot\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \end{aligned}$$

Reversing the $(\cdot)^\star$-translation step by step yields

$$\begin{aligned} &r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\bot\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \\ =&r \wedge \nabla\{\neg q \wedge \nabla\{\Box\bot, \Box p\}, \top\} \\ =&r \wedge \nabla\{\neg q \wedge \Diamond\Box\bot \wedge \Diamond\Box p \wedge \Box(\Box\bot \vee \Box p), \top\} \\ =&r \wedge \Diamond(\neg q \wedge \Diamond\Box\bot \wedge \Diamond\Box p \wedge \Box(\Box\bot \vee \Box p)) \end{aligned}$$

Lastly, undoing the Sep-function and negating yields a syntactically open equivalent, negative in p:

$$\neg\mathbf{i} \vee \Box(q \vee \Box\Diamond\top \vee \Box\Diamond\neg p \vee \Diamond(\Diamond\top \wedge \Diamond\neg p))$$

Admittedly this equivalent could be simpler. Indeed, as noted in example 6.1.1, it is in fact equivalent to $\neg\mathbf{i} \vee \Box(\Box\Diamond\top \vee \Diamond\Diamond\neg p \vee q)$. The introduction of the subformula $\Box\Diamond\neg p$ is worrying, as this quantifier pattern is often the cause of SQEMA's failure. However, for the input formula in example 6.1.1 this causes no problem, as the reader can check. More sophisticated strategies for undoing $(\cdot)^\star$ should be able to minimize this problem — all that we are doing at the moment is applying the definition in reverse. $\lhd$

Example 6.3.12 In example 6.1.7 the monotonicity of the formula

$$(\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$$

was used in an application of the semantic Ackermann rule with replacement. As this formula is not separately monotone, the method presented in this section will not suffice to compute an equivalent negative in p in this case. Indeed,

$$(\Box_2\neg t \vee \neg p \vee \Diamond_1 p) \wedge \Diamond_1\Box_2(\Box_2\neg t \vee \neg p)$$

is not monotone in p. $\lhd$

These examples illustrate that monotone sequents in SSQEMA^r-executions often satisfy the stronger property of separated monotonicity. Syntactically correct equivalents of these sequents can be computed by using the method of simulation quantifiers presented in this section. However, as example 6.3.12 illustrates, SSQEMA^r-executions may give rise to sequents which are monotone but not separately monotone. To compute syntactically correct equivalents for these, stronger methods will have to be considered.

6.4 Negative equivalents for propositionally monotone formulae

The previous section concluded with the observation that the formula $(\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$, which is downward monotone in p, is not separately monotone in p. The methods of that section were hence insufficient to deal with such formulae. Note however, that the formula $(\Box^{-1}\neg r \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg r \vee \neg p)$, obtained by substituting the new propositional variable r for the nominal $\mathbf{i}$ *is* downward monotone in p. So, although the monotonicity of the formula does depend on the interpretation of $\Diamond$ and $\Diamond^{-1}$ as inverses, it does not depend on the interpretation of $\mathbf{i}$ as a singleton. Let us formalize this idea, by giving the following analogues of definitions 6.3.1 and 6.3.2.

Definition 6.4.1 Given a formula $\varphi \in \mathcal{L}_r^n$, the *propositional separation* of φ, denoted $\mathsf{PSep}(\varphi)$, is the formula obtained by uniformly substituting a fresh propositional variable for every nominal occurring in φ.

Definition 6.4.2 An $\mathcal{L}_r^n$-formula φ is *propositionally upward monotone* in a propositional variable p if $\mathsf{PSep}(\varphi)$ is upwards monotone in p. The notion of *propositional downward monotonicity* is defined similarly.

Clearly any separately monotone formula is propositionally monotone, and any propositionally monotone formula is monotone in the ordinary sense. In this section we develop methods for computing syntactically correct equivalents for syntactically closed (open) $\mathcal{L}_r^n$-formulae which are propositionally upward (downward) monotone in given propositional variables. These methods will again be based on suitably adapted bisimulation quantifiers, but will be considerably more involved than those of the previous section, due to the need to account adequately for the interaction between the modalities and their inverses.

6.4.1 Disjunctive forms for syntactically closed $\mathcal{L}_r$-formulae

An $\mathcal{L}_r$-formula is syntactically closed (open) if its is syntactically closed (open) when regarded as a $\mathcal{L}_r^n$-formula. In this subsection we introduce disjunctive forms for syntactically closed $\mathcal{L}_r$-formulae and study some of their properties. Define $\bigcirc(\{\varphi_1, \ldots, \varphi_n\}|\{\psi_1, \ldots, \psi_m\})$ as a shorthand for

$$\bigwedge_{i=1}^{n} \Diamond\varphi_i \wedge \Box \bigvee_{i=1}^{n} \varphi_i \wedge \bigwedge_{i=1}^{m} \Diamond^{-1}\psi_i.$$

The notation '$\bigcirc$' is intended to be suggestive of the R-neighbourhood *around* a point in a model. We will often omit the curly brackets and simply write $\bigcirc(\varphi_1, \ldots, \varphi_n | \psi_1, \ldots, \psi_m)$ for $\bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})$.

Definition 6.4.3 An $\mathcal{L}_r$-formula is in *disjunctive form* if it can be obtained using the following recursion:

$$\varphi ::= \top \mid \bot \mid \chi \mid \varphi \vee \psi \mid \chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})$$

where χ is a, possibly empty, conjunction of literals, and $m, n \in \{0, 1, 2, \ldots\}$. As before, the empty conjunction and disjunction are identified with $\top$ and $\bot$, respectively.

Convention 6.4.4 *Henceforth we will use the symbol χ only to denote conjunctions of literals.*

By removing disjunction from the above recursion we obtain the class of *disjunction free formulae in disjunctive form* (or the DFDF-formulae, for short). Formally,

$$\varphi ::= \top \mid \bot \mid \chi \mid \chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})$$

where χ, m and n are as before.

A formula is in *strict disjunctive form* (or is an SDF-formula, for short) if it is a disjunction of DFDF-formulae.

We further define $\mathsf{Bool}(\varphi)$, $\mathsf{Fut}(\varphi)$ and $\mathsf{Past}(\varphi)$ on DFDF-formulae, φ, as follows: $\mathsf{Bool}(\top) = \top$; $\mathsf{Bool}(\bot) = \bot$; $\mathsf{Bool}(\chi) = \chi$; $\mathsf{Bool}(\chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})) = \chi$. Let $\mathsf{Fut}(\top) = \emptyset$; $\mathsf{Fut}(\bot) = \emptyset$; $\mathsf{Fut}(\chi) = \emptyset$; $\mathsf{Fut}(\chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})) = \{\varphi_1, \ldots, \varphi_n\}$. Lastly, let $\mathsf{Past}(\top) = \emptyset$; $\mathsf{Past}(\bot) = \emptyset$; $\mathsf{Past}(\chi) = \emptyset$; $\mathsf{Past}(\chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\} | \{\psi_1, \ldots, \psi_m\})) = \{\psi_1, \ldots, \psi_m\}$.

Definition 6.4.5 Let Θ be a finite set of propositional variables. We define inductively what it means for a DFDF to be Θ-*full*: $\top$ and $\bot$ are both Θ-full; a conjunction of literals χ is Θ-full if each literal appearing in χ is a Θ-literal and for each $p \in \Theta$, a p-literal appears in χ. Lastly $\chi \wedge \bigcirc(\Phi | \Psi)$ is Θ-full if χ as well as each formula in $\Phi \cup \Psi$ is Θ-full. A DFDF-formula φ is *full* if it is $\mathsf{PROP}(\varphi)$-full.

Definition 6.4.6 The notion of *modal depth* is adapted to formulae in disjunctive form in the natural way:

$$\begin{aligned} depth(\bot) = depth(\top) = depth(\chi) &= 0 \\ depth(\varphi \vee \psi) &= \max(depth(\varphi), depth(\psi)) \\ depth(\chi \wedge \bigcirc(\Phi|\Psi)) &= 1 + \max\{depth(\gamma) \mid \gamma \in \Phi \cup \Psi\} \end{aligned}$$

We recursively define what it means for a DFDF-formula to be of *uniform depth* k, for some $k \in \omega$: $\top$, $\bot$ and χ are of uniform depth 0; $\varphi = \chi \wedge \bigcirc(\Phi|\Psi)$ is of uniform depth k if each element of $\Phi \cup \Psi$ is of uniform depth $k - 1$. A DFDF-formula, φ, is *of uniform depth* if it is of uniform depth $depth(\varphi)$.

Example 6.4.7 The DFDF-formula

$$p \wedge \neg q \wedge \bigcirc(\{\neg p \wedge \neg q \wedge \bigcirc(\emptyset|\{\top\}), \top\}|\emptyset)$$

is not of uniform depth. It is, however, equivalent to the following disjunction of DFDF-formulae of uniform depth:

$$\begin{array}{ll} & p \wedge \neg q \wedge \bigcirc(\{\neg p \wedge \neg q \wedge \bigcirc(\emptyset|\{\top\}), \top \wedge \bigcirc(\emptyset|\emptyset)\}|\emptyset) \\ \vee & p \wedge \neg q \wedge \bigcirc(\{\neg p \wedge \neg q \wedge \bigcirc(\emptyset|\{\top\}), \top \wedge \bigcirc(\{\top\}|\emptyset)\}|\emptyset) \end{array}$$

$\lhd$

Definition 6.4.8 A DFDF-formula is said to be *complete* if it is full, of uniform depth, and can be obtained from the recursion

$$\varphi ::= \chi \mid \chi \wedge \bigcirc(\{\varphi_1, \ldots, \varphi_n\}|\{\psi_1, \ldots, \psi_m\}).$$

An SDF-formula is *complete* if it is a disjunction of complete DFDF-formulae.

Thus a complete DFDF-formula may not contain the symbols $\top$ or $\bot$. Note that the different disjuncts of an SDF-formula need not be full with respect to the same set of propositional variables or of the same uniform depth.

A complete DFDF of depth k can be seen as a type of *characteristic formula* of a world in a model describing, up to syntactically closed simulation, the neighbourhood of that world up to k steps away. For more on characteristic formulae, see [GO06]. Indeed, we can formalize this idea as follows:

Definition 6.4.9 Let $(\mathcal{M}, m) = ((W, R, V), m)$ be a pointed model, $k \in \omega$ and Θ a finite set of propositional variables. The *characteristic formula* of $(\mathcal{M}, m)$ over Θ and of modal depth k, denoted $Char^k_\Theta(\mathcal{M}, m)$ is defined as follows, by induction on k:

1. $Char^0_\Theta(\mathcal{M}, m)$ is a conjunction of literals χ such that for each $p \in \Theta$, p is a conjunct of χ iff $(\mathcal{M}, m) \Vdash p$ and $\neg p$ is a conjunct of χ iff $(\mathcal{M}, m) \Vdash \neg p$.
2. $Char^k_\Theta(\mathcal{M}, m) = \chi \wedge \bigcirc(\Phi \mid \Psi)$ where χ is defined as in item 1, and $\Phi = \{Char^{k-1}_\Theta(\mathcal{M}, u) \mid Rmu\}$ and $\Psi = \{Char^{k-1}_\Theta(\mathcal{M}, v) \mid Rvm\}$.

Note that the sets Φ and Ψ in clause 2 are finite, and that this is the case even if we do not work modulo semantic equivalence. Because of the asymmetry in the way that the arguments of $\bigcirc(\cdot \mid \cdot)$ are treated, definition 6.4.9 is in fact too weak to be regarded as defining a proper characteristic formula in the spirit of [GO06]. If we had wanted to obtain a characteristic formula in that sense, we should have added the additional conjunct $\Box^{-1} \bigvee \Psi$ to the definition of $\bigcirc(\Phi \mid \Psi)$. The reason for not having done so, is of course that we want to restrict attention to syntactically closed (and via negation, syntactically open) formulae. The following lemma is immediate from definition 6.4.9.

Lemma 6.4.10 *$(\mathcal{M}, m) \Vdash Char^k_\Theta(\mathcal{M}, m)$, for any pointed model $(\mathcal{M}, m)$, finite set of propositional variables Θ and $k \in \omega$.*

Lemma 6.4.11 *Every syntactically closed $\mathcal{L}_r$-formula φ can be effectively and equivalently rewritten as a complete SDF-formula.*

Proof. Let $\varphi \in \mathcal{L}_r$ be syntactically closed and in negation normal form. We first rewrite φ equivalently as a formula in disjunctive form, using the translation $(\cdot)^\star$, recursively defined by

$$\begin{aligned}
\top^\star &= \top \\
\bot^\star &= \bot \\
p^\star &= p \\
(\varphi \vee \psi)^\star &= \varphi^\star \vee \psi^\star \\
(\Diamond\varphi)^\star &= \bigcirc(\varphi^\star, \top \mid \emptyset) \\
(\Diamond^{-1}\varphi)^\star &= \bigcirc(\emptyset \mid \varphi^\star) \vee \bigcirc(\top \mid \varphi^\star) \\
(\Box\varphi)^\star &= \bigcirc(\emptyset \mid \emptyset) \vee \bigcirc(\varphi^\star \mid \emptyset).
\end{aligned}$$

The case for conjunction is more complicated. Consider a formula of the form $\bigwedge S$. If S is such that $S = S' \cup \{\top\}$, $S = S' \cup \{\bot\}$ or $S = S' \cup \{\varphi \vee \psi\}$ we translate as follows:

$$\begin{aligned}
(\bigwedge(S' \cup \{\top\}))^\star &= (\bigwedge S')^\star \\
(\bigwedge(S' \cup \{\bot\}))^\star &= \bot \\
(\bigwedge(S' \cup \{\varphi \vee \psi\}))^\star &= (\bigwedge(S' \cup \{\varphi\}))^\star \vee (\bigwedge(S' \cup \{\psi\}))^\star.
\end{aligned}$$

If S does not contain $\top$, $\bot$ or a disjunction, it means that every formula in S is either a literal or a formula of the form $\Diamond\psi$, $\Box\psi$, or $\Diamond^{-1}\psi$. We form the following sets:

$$\begin{aligned}
S_\Diamond &= \{\psi \mid \Diamond\psi \in S\} \\
S_\Box &= \{\psi \mid \Box\psi \in S\} \\
S_{\Diamond^{-1}} &= \{\psi \mid \Diamond^{-1}\psi \in S\}
\end{aligned}$$

Lastly, let S_{lit} be the subset of all literals in S. If $S_\Diamond \neq \emptyset$ let

$$(\bigwedge S)^\star = \bigwedge S_{lit} \wedge \bigcirc(\{(\varphi \wedge \bigwedge S_\Box)^\star \mid \varphi \in S_\Diamond \cup \{\top\}\} \mid \{\psi^\star \mid \psi \in S_{\Diamond^{-1}}\}).$$

If, on the other hand, $S_\Diamond = \emptyset$, then let

$$(\bigwedge S)^\star = (\bigwedge S_{lit} \wedge \bigcirc(\emptyset \mid \{\psi^\star \mid \psi \in S_{\Diamond^{-1}}\})) \vee (\bigwedge S_{lit} \wedge \bigcirc(\{(\bigwedge S_\Box)^\star\} \mid \{\psi^\star \mid \psi \in S_{\Diamond^{-1}}\})).$$

Thus the translation $(\cdot)^\star$ yields an equivalent formula in disjunctive form. The following equivalences can now be used to convert this formula into an equivalent complete SD-formula:

$$\varphi \equiv \varphi \wedge \bigcirc(\emptyset \mid \emptyset) \vee \bigcirc(\top \mid \emptyset) \tag{6.1}$$

$$\varphi \equiv (\varphi \wedge p) \vee (\varphi \wedge \neg p) \tag{6.2}$$

$$\begin{aligned} & \chi \wedge \bigcirc(\{\alpha \vee \beta\} \cup \Phi|\Psi) \\ \equiv\ & (\chi \wedge \bigcirc(\{\alpha\} \cup \Phi|\Psi)) \vee (\chi \wedge \bigcirc(\{\beta\} \cup \Phi|\Psi)) \vee (\chi \wedge \bigcirc(\{\alpha, \beta\} \cup \Phi|\Psi)) \end{aligned} \tag{6.3}$$

$$\begin{aligned} & \chi \wedge \bigcirc(\Phi|\{\alpha \vee \beta\} \cup \Psi) \\ \equiv\ & (\chi \wedge \bigcirc(\Phi|\{\alpha\} \cup \Psi)) \vee (\chi \wedge \bigcirc(\Phi|\{\beta\} \cup \Psi)) \end{aligned} \tag{6.4}$$

QED

Example 6.4.12 We illustrate lemma 6.4.11 by converting the syntactically open $\mathcal{L}_r$-formula $p \wedge \neg q \wedge \Diamond q \wedge \Box p \wedge \Diamond^{-1}(\neg q \wedge \neg p)$ into a complete SDF-formula:

$$\begin{aligned} & (p \wedge \neg q \wedge \Diamond q \wedge \Box p \wedge \Diamond^{-1}(\neg q \wedge \neg p))^\star \\ =\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), p\} \mid \{\neg q \wedge \neg p\}) \end{aligned}$$

The latter formula is an SDF-formula, but it is not complete. We now apply equivalences 6.2 and 6.3 to make it complete:

$$\begin{aligned} & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), p\} \mid \{\neg q \wedge \neg p\}) \\ \equiv\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), (p \wedge q) \vee (p \wedge \neg q)\} \mid \{\neg q \wedge \neg p\}) \\ \equiv\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), (p \wedge q)\} \mid \{\neg q \wedge \neg p\}) \\ \vee\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), (p \wedge \neg q)\} \mid \{\neg q \wedge \neg p\}) \\ \vee\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), (p \wedge q), (p \wedge \neg q)\} \mid \{\neg q \wedge \neg p\}). \end{aligned}$$

This formula may further be simplified using the commutativity of $\wedge$, to obtain:

$$\begin{aligned} & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p)\} \mid \{\neg q \wedge \neg p\}) \\ \vee\ & p \wedge \neg q \wedge \bigcirc(\{(q \wedge p), (p \wedge \neg q)\} \mid \{\neg q \wedge \neg p\}). \end{aligned}$$

$\lhd$

6.4.2 Coherent formulae and standard models

Next we define a relation between formulae in disjunctive form. The significance of this definition is given in proposition 6.4.15 below. Continuing with the characteristic formulae-intuition for DFDF's, $\varphi \preceq \psi$ means that φ and ψ describe the same world, but that φ is possibly *more specific*, or *more detailed*, than ψ. This relation, together with a weaker version of it, will also allow us to syntactically characterize the satisfiable complete DFDF-formulae. The obtained syntactic criterion will be called (weak) *coherency*. This characterization will be needed when we define simulation quantifiers to act on these formulae. We will further define so-called *standard models* for (weakly) coherent DFDF-formulae which characterize the models of these formulae modulo certain syntactically closed simulations. We will use the these standard models to prove an analogue of proposition 6.3.9.

Definition 6.4.13 Let Θ be a possibly empty set of propositional variables. Given two conjunctions of literals χ_1 and χ_2, we write $\chi_1 \subseteq_\Theta \chi_2$ if

1. each positive Θ-literal in χ_1 also appears in χ_2,
2. each non-Θ-literal in χ_1 also appears in χ_2.

Suppose $depth(\psi) \leq depth(\varphi)$. We define the relations $\varphi \preceq_\Theta \psi$ and $\varphi \leq_\Theta \psi$ by induction on φ. We start with $\varphi \preceq_\Theta \psi$:

1. $\top \preceq_\Theta \psi$ if and only if $\psi = \top$;
2. $\bot \preceq_\Theta \psi$ always (i.e. if $\psi = \top$ or $\psi = \bot$ or $\psi = \chi$);
3. $\chi \preceq_\Theta \psi$ if and only if
   (a) $\psi = \top$, or
   (b) $\psi = \bot$ and χ is inconsistent, or
   (c) $\psi = \chi'$ with $\chi' \subseteq_\Theta \chi$;
4. For φ of the form $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1))$, we specify that
   (a) $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \preceq_\Theta \top$;
   (b) $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \preceq_\Theta \bot$ whenever χ_1 is inconsistent;
   (c) $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \preceq_\Theta \chi_2$ whenever $\chi_2 \subseteq_\Theta \chi_1$;
   (d) $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \preceq_\Theta (\chi_2 \wedge \bigcirc(\Phi_2|\Psi_2))$ if
      i. $\chi_2 \subseteq_\Theta \chi_1$, and
      ii. for each $\varphi_1 \in \Phi_1$ there is a $\varphi_2 \in \Phi_2$ such that $\varphi_1 \preceq_\Theta \varphi_2$,
      iii. for each $\varphi_2 \in \Phi_2$ there is a $\varphi_1 \in \Phi_1$ such that $\varphi_1 \preceq_\Theta \varphi_2$, and
      iv. for each $\psi_2 \in \Psi_2$, there is a $\psi_1 \in \Psi_1$ such that $\psi_1 \preceq \psi_2$.

The relation $\varphi \leq_\Theta \psi$ is defined by replacing '$\preceq_\Theta$' everywhere in the above with '$\leq_\Theta$', and by replacing clause 4(d) with the clause:

4. (e) $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \leq_\Theta (\chi_2 \wedge \bigcirc(\Phi_2|\Psi_2))$ if
      i. $\chi_2 \subseteq_\Theta \chi_1$, and
      ii. for each $\varphi_1 \in \Phi_1$ there is a $\varphi_2 \in \Phi_2$ such that $\varphi_1 \leq_\Theta \varphi_2$, and
      iii. for each $\varphi_2 \in \Phi_2$ there is a $\varphi_1 \in \Phi_1$ such that $\varphi_1 \leq_\Theta \varphi_2$.

We will write $\subseteq$, $\leq$ and $\preceq$ for $\subseteq_\emptyset$, $\leq_\emptyset$ and $\preceq_\emptyset$, respectively.

Note that the difference between the relations $\preceq_\Theta$ and $\leq_\Theta$ is essentially confined to the treatment of elements of second argument of the $\bigcirc(\cdot \mid \cdot)$-construct.

Example 6.4.14 Here are some example illustrating the $\preceq$ and $\leq$-relations:

$$[p \wedge \neg q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q, p \wedge q)] \preceq [p \wedge \neg q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q)]$$

and

$$[p \wedge \neg q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q, p \wedge q)] \preceq [p \wedge \bigcirc(q, \neg q \mid \neg q)],$$

but

$$[p \wedge \neg q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q, p \wedge q)] \not\preceq [p \wedge \neg q \wedge \bigcirc(\neg p \wedge \neg q \mid p \wedge q)].$$

Lastly

$$[p \wedge q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q, p \wedge q)] \preceq_{\{q\}} [p \wedge \neg q \wedge \bigcirc(\neg p \wedge \neg q, q \mid q)]$$

and

$$[p \wedge q \wedge \bigcirc(\neg p \wedge q, \neg p \wedge \neg q \mid \neg p \wedge \neg q)] \leq_{\{q\}} [p \wedge \neg q \wedge \bigcirc(\neg p \wedge \neg q, q \mid q)].$$

Note the asymmetry in the way in which the first and second arguments of the $\bigcirc$-operator are treated. $\lhd$

Proposition 6.4.15 *For any DFDF-formulae φ and ψ,*

1. *if $\varphi \preceq \psi$, then $\Vdash \varphi \to \psi$, and*
2. *if φ is complete, $\mathsf{PROP}(\psi) \subseteq \mathsf{PROP}(\varphi)$ and $\varphi \not\preceq \psi$, then $\Vdash \varphi \wedge \psi \to \bot$.*

Proof. We proceed by induction of φ. The base cases are trivial for both claims. (Note that, in the second case, when φ is complete, the cases for $\varphi = \bot$ and $\varphi = \top$ do not occur.) So suppose that φ is of the form $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1))$. The cases when ψ is $\top$, $\bot$ or χ_2 are again trivial for both claims. Hence, suppose that ψ is of the form $(\chi_2 \wedge \bigcirc(\Phi_2|\Psi_2))$. For the sake of the first claim, suppose that $(\chi_1 \wedge \bigcirc(\Phi_1|\Psi_1)) \preceq (\chi_2 \wedge \bigcirc(\Phi_2|\Psi_2))$, and let $(\mathcal{M}, m)$ be any model such that $(\mathcal{M}, m) \Vdash \varphi$ — the claim clearly follows if no such model exists. Since $\chi_2 \subseteq \chi_1$, $(\mathcal{M}, m) \Vdash \chi_2$.

Let v be any successor of m in $\mathcal{M}$. To see that v satisfies some member of Φ_2, we note that v satisfies some member of Φ_1, say φ_1'. But then, since $\varphi \preceq \psi$, there is some member of Φ_2, say φ_2', such that $\varphi_1' \preceq \varphi_2'$, and hence, by the inductive hypothesis, $\Vdash \varphi_1' \to \varphi_2'$, and hence $(\mathcal{M}, v) \Vdash \varphi_2'$.

Now, let φ_2' be any member of Φ_2. To see that there is some successor of m in $\mathcal{M}$ which makes φ_2' true, we note that there is some element of Φ_1, say φ_1', such that $\varphi_1' \preceq \varphi_2'$. But there is some successor, say v, of m in $\mathcal{M}$ such that $(\mathcal{M}, v) \Vdash \varphi_1'$. Hence, by the inductive hypothesis, $\Vdash \varphi_1' \to \varphi_2'$, and hence $(\mathcal{M}, v) \Vdash \varphi_2'$.

A symmetric argument establishes the fact that every member of Ψ_2 is satisfied at some predecessor of m. We conclude that $(\mathcal{M}, m) \Vdash \psi$, and, since $(\mathcal{M}, m)$ was arbitrary, that $\Vdash \varphi \to \psi$.

As for the second claim of the lemma, suppose that $\varphi \not\preceq \psi$. The case when $\psi = \top$ does not occur, while the case for $\psi = \bot$ is clear. Suppose that ψ is of the form χ_2. Then it must be the case that $\chi_2 \not\subseteq \chi_1$. Since χ_1 is $\mathsf{PROP}(\varphi)$-full and $\mathsf{PROP}(\psi) \subseteq \mathsf{PROP}(\varphi)$, it must be the case that for some $p \in \mathsf{PROP}(\psi)$, χ_1 and χ_2 contain complimentary p-literals. Clearly φ and ψ cannot be simultaneously satisfiable.

Lastly suppose that ψ is of the form $\chi_2 \wedge \bigcirc(\Phi_2|\Psi_2)$. If $\chi_2 \not\subseteq \chi_1$, then the claim follows as above. Otherwise, there are two possibilities. Firstly, suppose there is some $\varphi_1 \in \Phi_1$ such that, for each $\varphi_2 \in \Phi_2$, $\varphi_1 \not\lesssim \varphi_2$, i.e., by the inductive hypothesis, $\Vdash \varphi_1 \wedge \varphi_2 \to \bot$. Now, on any model $(\mathcal{M}, m)$ such that $(\mathcal{M}, m) \Vdash \varphi$, m must have a successor, say v, in $\mathcal{M}$ making φ_1 true. But then, since v cannot also satisfy any member of Φ_2, we must have $(\mathcal{M}, m) \Vdash \neg\psi$. The second possibility, namely that there is some $\varphi_2 \in \Phi_2$ such that, for each $\varphi_1 \in \Phi_1$, $\varphi_1 \not\lesssim \varphi_2$, is analogous. QED

The notion of the coherency of a DFDF-formula is defined next. As will be seen further, this in fact represents a syntactic characterization of the satisfiable complete DFDF-formulae.

Definition 6.4.16 We define by induction what it means for a DFDF-formula of uniform depth to be (weakly) *coherent*. All coherent formulae are also weakly coherent. A formula which is not weakly coherent is *incoherent*.

1. $\top$ is coherent;
2. $\bot$ is incoherent;
3. a conjunction of literals χ is coherent if and only if it is consistent;
4. $\varphi = \chi \wedge \bigcirc(\Phi|\Psi)$ is coherent if and only if
   (a) χ is consistent,
   (b) each element of $\Phi \cup \Psi$ is coherent, and
   (c) for each $\psi' \in \Psi$, $\varphi \preceq \delta$ for some $\delta \in \mathsf{Fut}(\psi')$.
5. $\varphi = \chi \wedge \bigcirc(\Phi|\Psi)$ is weakly coherent if and only if
   (a) χ is consistent,
   (b) each element of $\Phi \cup \Psi$ is weakly coherent, and
   (c) for each $\psi' \in \Psi$, $\varphi \leq \delta$ for some $\delta \in \mathsf{Fut}(\psi')$.

An SDF-formula is (weakly) *coherent* if it is a disjunction of (weakly) coherent DFDF-formulae.

The intuition behind clauses 4(c) and 5(c) above is that the characteristic formula of each predecessor of a world m must list (a shallower version of) the characteristic formula of m as a possible successor.

Example 6.4.17 The formula $p \wedge \bigcirc(p, \neg p \mid \neg p \wedge \bigcirc(p \wedge \bigcirc(p, \neg p \mid \neg p) \mid \emptyset))$ is coherent, while the formula $p \wedge \bigcirc(p, \neg p \mid \neg p \wedge \bigcirc(p \wedge \bigcirc(p, \neg p \mid p) \mid \emptyset))$ is only weakly coherent. The formula $p \wedge \bigcirc(p, \neg p \mid \neg p \wedge \bigcirc(\neg p \mid \emptyset))$ is incoherent. $\lhd$

More instances of coherent formulae are given by the following lemma:

Lemma 6.4.18 *For any pointed model $(\mathcal{M}, m)$ set of propositional variables Θ, and $k \in \omega$, the formula $Char^k_\Theta(\mathcal{M}, m)$ is a complete and coherent DFDF-formula.*

Proof. By induction on k. QED

Standard models for coherent formulae

For each complete and coherent DFDF-formula φ we wish to define a pointed model which satisfies φ, and which, up to syntactically closed $depth(\varphi)$-simulation, is characteristic of all models satisfying φ. We begin by defining some operations on models.

Let $\{\mathcal{M}_i = (W_i, R_i, V_i)\}_{i\in I}$ be a family of models, the domains of which have been made pairwise disjoint, e.g. by some suitable indexing. Recall that the *disjoint union* of the family $\{\mathcal{M}_i = (W_i, R_i, V_i)\}_{i\in I}$, denoted $\biguplus\{\mathcal{M}_i\}_{i\in I}$, is the model with (W, R, V) with $W = \bigcup_{i\in I} W_i$, $R = \bigcup_{i\in I} R_i$ and $V(p) = \bigcup_{i\in I} V_i(p)$ for each propositional variable p.

The following construction will be used often: Let $\{(\mathcal{M}_i, m_i)\}_{i\in I}$ and $\{(\mathcal{N}_j, n_j)\}_{j\in J}$ be two families of pointed models, w a point not in the domain of any $\mathcal{M}_i$ or $\mathcal{N}_j$, and χ a possibly infinite consistent conjunction of literals. Then

$$\mathcal{COMB}(\{(\mathcal{M}_i, m_i)\}_{i\in I}, \{(\mathcal{N}_j, n_j)\}_{j\in J}, w, \chi) = (W, R, V)$$

is the model constructed by adding to the disjoint union of all members of $\{(\mathcal{M}_i, m_i)\}_{i\in I}$ and $\{(\mathcal{N}_j, n_j)\}_{j\in J}$ the new point w, and making each m_i, $i \in I$, a successor of w and each n_j, $j \in J$, a predecessor of w. The valuation is extended so as to make χ true at w, and all proposition letters in the language not appearing in χ are declared false at w. More formally, let

$$\mathcal{M}_0 = \left(\biguplus\{(\mathcal{M}_i, m_i)\}_{i\in I}\right) \uplus \left(\biguplus\{(\mathcal{N}_j, n_j)\}_{j\in J}\right),$$

and let $W = W^{\mathcal{M}_0} \cup \{w\}$, $R = R^{\mathcal{M}_0} \cup \{(w, m_i) \mid i \in I\} \cup \{(n_j, w) \mid j \in J\}$, and V is such that $V(p) = V^{\mathcal{M}_0}(p) \cup \{w\}$ if p occurs positively in χ, and $V(p) = V^{\mathcal{M}_0}(p)$, otherwise.

Note that the $\mathcal{COMB}$-construction does *not* in general preserve the truth of (syntactically closed) formulae at the points n_j, $j \in J$, since these points gain new successors in the construction. We will further develop conditions sufficient to guarantee the preservation of truth, up to certain modal depths, under this construction. The proof of the following lemma is routine.

Lemma 6.4.19 *Let $\{(\mathcal{M}_i, m_i)\}_{i\in I}$ and $\{(\mathcal{N}_j, n_j)\}_{j\in J}$ be two families of pointed models, w a point not in the domain of any $\mathcal{M}_i$ or $\mathcal{N}_j$, χ a consistent conjunction of literals, and $k \in \omega$. Then*

1. *for each $i \in I$,*

$$(\mathcal{M}_i, m_i) \rightrightarrows^k_{SC(\emptyset)} (\mathcal{COMB}(\{(\mathcal{M}_i, m_i)\}_{i\in I}, \{(\mathcal{N}_j, n_j)\}_{j\in J}, w, \chi), m_i),$$

and

2. *for each $j \in J$, if there is a successor n'_j of n_j in $(\mathcal{N}_j, n_j)$ (i.e $R^{\mathcal{N}_j} n_j n'_j$) such that $(\mathcal{N}_j, n'_j) \rightrightarrows^{k-1}_{SC(\emptyset)} (\mathcal{COMB}(\{(\mathcal{M}_i, m_i)\}_{i\in I}, \{(\mathcal{N}_j, n_j)\}_{j\in J}, w, \chi), w)$, then*

$$(\mathcal{N}_j, n_j) \rightrightarrows^k_{SC(\emptyset)} (\mathcal{COMB}(\{(\mathcal{M}_i, m_i)\}_{i\in I}, \{(\mathcal{N}_j, n_j)\}_{j\in J}, w, \chi), n_j).$$

We next define the pointed *standard model*, $(\mathcal{SM}(\varphi), w_\varphi)$, of a coherent, complete DFDF-formula, φ. The standard model of χ consists of a single, irreflexive point at which all proposition letters in the language, other than those appearing positively in χ, are false. Formally, let $(\mathcal{SM}(\chi), w_\chi) = ((W, R, V), w)$, with $W = \{w\}$, $R = \emptyset$, and $V(p) = \{w\}$ for all proposition letters p such that p is a conjunct if χ, and $V(p) = \emptyset$, otherwise.

Now suppose $\varphi = \chi \wedge \bigcirc(\Phi|\Psi)$, and that the standard models of all the elements of $\Phi \cup \Psi$ have already been defined. (Note that, by the coherency of φ, each element of $\Phi \cup \Psi$ is coherent.) We set

$$(\mathcal{SM}(\varphi), w_\varphi) = (\mathcal{COMB}(\{(\mathcal{SM}(\alpha), w_\alpha) \mid \alpha \in \Phi\}, \{(\mathcal{SM}(\beta), w_\beta) \mid \beta \in \Psi\}, w_\varphi, \chi), w_\varphi).$$

The next two lemmas describe, respectively, a relation between the standard models of $\preceq_\Theta$-related formulae, and some internal relations within standard models.

Lemma 6.4.20 *For all complete and coherent DFDF-formulae φ and ψ and sets of propositional variables Θ such that $\varphi \preceq_\Theta \varphi'$, it holds that*

$$(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{depth(\varphi')}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_\varphi).$$

Proof. Let $\varphi = \chi \wedge \bigcirc(\Phi \mid \Psi)$ and $\varphi' = \chi' \wedge \bigcirc(\Phi' \mid \Psi')$ be as in the formulation of the lemma. We show that, $(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_\varphi)$ for each $k \leq depth(\varphi')$, proceeding by induction on k. We have to verify all clauses of definition 6.2.3. The base case for $k = 0$ is easy to verify. Now let $0 < k$ and suppose that the claim holds for all natural numbers less than k. Since $\varphi \preceq_\Theta \varphi'$, we have $\chi' \subseteq_\Theta \chi$, whence the local harmony clause. For every $\varphi'_0 \in \Phi'$ there is a $\varphi_0 \in \Phi$ such that $\varphi_0 \preceq_\Theta \varphi'_0$, and hence, by the inductive hypothesis, such that

$$(\mathcal{SM}(\varphi'_0), w_{\varphi'_0}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi_0), w_{\varphi_0}). \tag{6.5}$$

Further, also by the inductive hypothesis, we have that

$$(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k-2}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_\varphi). \tag{6.6}$$

It follows from 6.5, 6.6 and the construction of $\mathcal{SM}(\varphi)$ and $\mathcal{SM}(\varphi')$ that

$$(\mathcal{SM}(\varphi'), w_{\varphi'_0}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0}). \tag{6.7}$$

Similarly we have that, for each $\varphi_0 \in \Phi$, there is a $\varphi'_0 \in \Phi'$ such that

$$(\mathcal{SM}(\varphi'), w_{\varphi'_0}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0}). \tag{6.8}$$

This deals with the forth clause. As for the asymmetric back clause, note that, for every $\psi'_0 \in \Psi'$ there is a $\psi_0 \in \Psi$ such that $\psi_0 \preceq_\Theta \psi'_0$. Hence, by the inductive hypothesis,

$$(\mathcal{SM}(\psi'_0), w_{\psi'_0}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\psi_0), w_{\psi_0}). \tag{6.9}$$

It follows from 6.9, 6.6 and the construction of $\mathcal{SM}(\varphi)$ and $\mathcal{SM}(\varphi')$ that

$$(\mathcal{SM}(\varphi'), w_{\psi_0'}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0}). \tag{6.10}$$

We conclude that

$$(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi}).$$

QED

Lemma 6.4.21 *For all complete and coherent DFDF-formulae $\varphi = \chi \wedge \bigcirc(\Phi \mid \Psi)$, it holds that*

1. $(\mathcal{SM}(\varphi_0), w_{\varphi_0}) \rightrightarrows^{depth(\varphi_0)}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0})$*, for all* $\varphi_0 \in \Phi$*, and*
2. $(\mathcal{SM}(\psi_0), w_{\psi_0}) \rightrightarrows^{depth(\psi_0)}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0})$*, for all* $\psi_0 \in \Psi$*.*

Proof. We show, proceeding by induction on k, that, for all $k < depth(\varphi)$,

1. $(\mathcal{SM}(\varphi_0), w_{\varphi_0}) \rightrightarrows^{k}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0})$ for all $\varphi_0 \in \Phi$, and
2. $(\mathcal{SM}(\psi_0), w_{\psi_0}) \rightrightarrows^{k}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0})$.

The base case for $k = 0$ is easy to verify. Indeed, for all $\varphi_0 \in \Phi$, the point w_{φ_0} has the same valuation both in $(\mathcal{SM}(\varphi_0)$ and $(\mathcal{SM}(\varphi))$, and hence $(\mathcal{SM}(\varphi_0), w_{\varphi_0}) \rightrightarrows^{0}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0})$. The case for $\psi_0 \in \Psi$ is similar. If $\varphi \preceq_{\Theta} \varphi'$ the $\mathsf{Bool}(\varphi') \subseteq_{\Theta} \mathsf{Bool}(\varphi)$, and hence $(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{0}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi})$.

Now let $0 < k$, and suppose that the claim holds for all natural numbers less than k. Clause 1 of the lemma follows immediately by lemma 6.4.19 and the construction of $(\mathcal{SM}(\varphi), w_{\varphi})$. For the sake of clause 2, let $\psi_0 \in \Psi$. By the coherence of φ there is a $\gamma \in \mathsf{Fut}(\varphi_0)$ such that $\varphi \preceq_{\emptyset} \gamma$. (Notice that, since φ is complete, we have $\mathsf{PROP}(\gamma) = \mathsf{PROP}(\varphi_0) = \mathsf{PROP}(\varphi)$). By lemma 6.4.20 $(\mathcal{SM}(\gamma), w_{\gamma}) \rightrightarrows^{k-1}_{SC(\emptyset)(\mathsf{PROP}(\varphi),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi})$. We would be able to apply lemma 6.4.19 and conclude that $(\mathcal{SM}(\psi_0), w_{\psi_0}) \rightrightarrows^{k}_{SC(\emptyset)(\mathsf{PROP}(\varphi),\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0})$ if we were able to show that also $(\mathcal{SM}(\psi_0), w_{\gamma}) \rightrightarrows^{k-1}_{SC(\emptyset)(\mathsf{PROP}(\varphi),\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi})$. Since w_{γ} has exactly the same successors in $\mathcal{SM}(\gamma)$ as in $\mathcal{SM}(\psi_0)$, the latter would *not* be the case only if w_{γ} had gained a predecessor in passing from $\mathcal{SM}(\gamma)$ to $\mathcal{SM}(\psi_0)$ which is not similar to any predecessor of w_{φ} in $\mathcal{SM}(\varphi)$. But the only predecessor gained in this way by w_{γ} is w_{ψ_0}, which is also a predecessor of w_{φ}. The claim follows. QED

Lemma 6.4.22 $(\mathcal{SM}(\varphi), w_{\varphi}) \Vdash \varphi$*, for every complete and coherent DFDF-formula* φ*.*

Proof. We proceed by induction on φ. The base case, when φ is a conjunction of literals χ, is clear. Suppose $\varphi = \chi \wedge \bigcirc(\Phi \mid \Psi)$ and that $depth(\varphi) = k$. Then, by the inductive hypothesis, $(\mathcal{SM}(\varphi_0), w_{\varphi_0}) \Vdash \varphi_0$ for all $\varphi_0 \in \Phi$ and $(\mathcal{SM}(\psi_0), w_{\psi_0}) \Vdash \psi_0$ for all $\psi_0 \in \Phi$. By lemma 6.4.21, $(\mathcal{SM}(\varphi_0), w_{\varphi_0}) \rightrightarrows^{k-1}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\varphi_0})$ for all $\varphi_0 \in \Phi$ and $(\mathcal{SM}(\psi_0), w_{\psi_0}) \rightrightarrows^{k-1}_{SC(\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0})$ for all $\psi_0 \in \Phi$. Note that $depth(\gamma) = k - 1$ for each $\gamma \in \Phi \cup \Psi$. It follows that $(\mathcal{SM}(\varphi), w_{\varphi_0}) \Vdash \varphi_0$ for all $\varphi_0 \in \Phi$ and $(\mathcal{SM}(\psi), w_{\psi_0}) \Vdash \psi_0$ for all $\psi_0 \in \Phi$. Moreover, $(\mathcal{SM}(\varphi), w_{\varphi}) \Vdash \chi$, by the construction of $(\mathcal{SM}(\varphi), w_{\varphi})$. We conclude that $(\mathcal{SM}(\varphi), w_{\varphi}) \Vdash \varphi$.
QED

Lemma 6.4.23 *Let φ be a complete and coherent DFDF-formula. Then for any model $(\mathcal{M}, m)$*

$$(\mathcal{M}, m) \Vdash \varphi \quad \textit{iff} \quad (\mathcal{SM}(\varphi), w_\varphi) \rightrightarrows^{depth(\varphi)}_{SC(\emptyset)(\mathsf{PROP}(\varphi),\emptyset)} (\mathcal{M}, m).$$

Proof. By induction on φ, with the aid of lemma 6.4.21. QED

Standard models for weakly coherent formulae

To be able to properly define simulation quantifiers for syntactically closed $\mathcal{L}_r$-formulae (in subsection 6.4.3), we will need a syntactic characterization of the satisfiable such formulae. In the previous section, where we dealt with separately monotone formulae, satisfiability could be determined already at the propositional level. For the propositionally monotone formulae of this section, however, the situation is more complicated.

Lemma 6.4.22 tells us that coherency is a sufficient condition for satisfiability of complete DFDF-formulae. As will be seen, a characterization can be obtained by weakening this condition to weak coherency.

To that aim, we define *weak standard models* $(\mathcal{WSM}(\varphi), w_\varphi)$ for weakly coherent complete DFDF-formulae φ, proceeding by induction on the modal depth of φ. For the base case, when φ is a conjunction of literals, let $(\mathcal{WSM}(\varphi), w_\varphi) = (\mathcal{SM}(\varphi), w_\varphi)$. Suppose φ is of the form $\chi \wedge \bigcirc(\Phi \mid \Psi)$, and that $(\mathcal{WSM}(\gamma), w_\gamma)$ has already been defined for each weakly coherent DFDF-formula γ with $PROP(\gamma) = \mathsf{PROP}(\varphi)$ and $depth(\gamma) \leq depth(\varphi)$. Let S be the set containing all these weak standard models. Then we define $(\mathcal{WSM}(\varphi), w_\varphi)$ to be the model

$$(\mathcal{COMB}(\{(\mathcal{WSM}(\varphi_0), w_{\varphi_0})\}_{\varphi_0 \in \Phi}, S, w_\varphi, \chi), w_\varphi).$$

In other words, $(\mathcal{WSM}(\varphi), w_\varphi)$ is defined like $(\mathcal{SM}(\varphi), w_\varphi)$, but we add as predecessors to w_φ all weak standard models of formulae of shallower depth. Note that, particularly, $(\mathcal{WSM}(\psi), w_\psi) \in S$ for all $\psi \in \Psi$. The proof of the next lemma illustrates the rationale behind this construction.

Lemma 6.4.24 *Let φ and ψ be two weakly coherent DFDF-formulae and Θ a set of propositional variables. If $\varphi \leq_\Theta \varphi'$, then $(\mathcal{WSM}(\varphi'), w_{\varphi'}) \rightrightarrows^{depth(\varphi')}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{WSM}(\varphi), w_\varphi)$.*

Proof. The proof is quite similar to that of lemma 6.4.20. Let $\varphi = \chi \wedge \bigcirc(\Phi \mid \Psi)$ and $\varphi' = \chi' \wedge \bigcirc(\Phi' \mid \Psi')$. We show that $(\mathcal{WSM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{WSM}(\varphi), w_\varphi)$ for all $0 \leq k \leq depth(\varphi')$, proceeding by induction on k. The base case, for $k = 0$, is clear. Suppose that $0 < k \leq depth(\varphi')$ and that the claim holds for all numbers less than k. We verify the relevant clauses of definition 6.2.3. Since $\chi' \subseteq_\Theta \chi$, the local harmony clauses follow. The forth clause is the same as that in the proof of lemma 6.4.20. We treat the non-reversive back clause.

Let v be any predecessor of $w_{\varphi'}$ in $\mathcal{WSM}(\varphi')$. Then, by the construction of $\mathcal{WSM}(\varphi')$, v is the root w_{ψ_0} of $(\mathcal{WSM}(\psi_0), w_{\psi_0})$ for some formula ψ_0 with $\mathsf{PROP}(\psi_0) = \mathsf{PROP}(\varphi')$ and $depth(\psi_0) < depth(\varphi')$. But, also by construction, the root of another copy of $(\mathcal{WSM}(\psi_0), w_{\psi_0})$

is a predecessor of w_φ in $(\mathcal{WSM}(\varphi), w_\varphi)$. Moreover, by the inductive hypothesis, we have that

$$(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k-2}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_\varphi).$$

Hence it follows by the construction of $(\mathcal{WSM}(\varphi), w_\varphi)$ and $(\mathcal{WSM}(\varphi'), w_{\varphi'})$ that

$$(\mathcal{SM}(\varphi'), w_{\psi_0}) \rightrightarrows^{k-1}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_{\psi_0}).$$

Thus the non-reversive back clause is verified. We conclude that

$$(\mathcal{SM}(\varphi'), w_{\varphi'}) \rightrightarrows^{k}_{SC(\Theta)(\mathsf{PROP}(\varphi'),\emptyset)} (\mathcal{SM}(\varphi), w_\varphi).$$

QED

Adapting the proofs of lemmas 6.4.21 and 6.4.22 in a similar way, we obtain the next two lemmas.

Lemma 6.4.25 *For all complete and weakly coherent DFDF-formulae $\varphi = \chi \wedge \bigcirc(\Phi \mid \Psi)$, it holds that*

1. $(\mathcal{WSM}(\varphi_0), w_{\varphi_0}) \rightrightarrows^{depth(\varphi_0)}_{SC(\emptyset)} (\mathcal{WSM}(\varphi), w_{\varphi_0})$ *for all $\varphi_0 \in \Phi$, and*
2. $(\mathcal{WSM}(\psi_0), w_{\psi_0}) \rightrightarrows^{depth(\psi_0)}_{SC(\emptyset)} (\mathcal{WSM}(\varphi), w_{\psi_0})$ *for all $\psi_0 \in \Psi$.*

Lemma 6.4.26 *For all complete and weakly coherent DFDF-formulae φ, it holds that*

$$(\mathcal{WSM}(\varphi), w_\varphi) \Vdash \varphi.$$

Theorem 6.4.27 *Let φ be a complete DFDF-formula, then φ is satisfiable iff it is weakly coherent.*

Proof. The right-to-left direction follows by lemma 6.4.26. Conversely, suppose that φ is an incoherent complete DFD-formula. We show, proceeding by induction on φ, that φ is not satisfiable. The base case, when φ is an inconsistent conjunction of literals, is clear. Suppose that φ is of the form $\chi \wedge \bigcirc(\Phi|\Psi)$. We check all possible cases, corresponding to the incoherency of φ:

CASE 1: χ is inconsistent. Then, since χ is a propositional contradiction, $\chi \wedge \bigcirc(\Phi|\Psi)$ is unsatisfiable.

CASE 2: Some member of $\Phi \cup \Psi$, say γ, is incoherent. Then, by the inductive hypothesis, γ is not satisfiable, hence φ is not satisfiable.

CASE 3: There exists a $\psi' \in \Psi$, such that $\varphi \not\lesssim \gamma$ for all $\gamma \in \mathsf{Fut}(\psi')$. Then, by lemma 6.4.15, $\Vdash \varphi \wedge \gamma \to \bot$ for all $\gamma \in \mathsf{Fut}(\psi')$. In other words, any point m satisfying φ must have a predecessor, each successor of which must satisfy the formula $\bigvee \mathsf{Fut}(\varphi')$, i.e. m must satisfy $\varphi \wedge \bigvee \mathsf{Fut}(\varphi')$, i.e. m must satisfy $\bot$. Hence, the unsatisfiability of φ. QED

6.4.3 A Lyndon-theorem for syntactically closed $\mathcal{L}_r$-formulae

The simulation quantifiers of definition 6.3.5 are easily adapted to $\mathcal{L}_r$-formulae in strict disjunctive form:

Definition 6.4.28 Let φ be an SDF-formula and $\overline{p}$ a vector of propositional variables. We define $\exists^+\overline{p}.\varphi$ inductively as follows:

$$\begin{array}{rcl} \exists^+\overline{p}.\bot &=& \bot \\ \exists^+\overline{p}.\top &=& \top \\ \exists^+\overline{p}.(\chi) &=& \chi' \end{array}$$

where χ' is $\bot$ when χ is inconsistent, or otherwise, when χ is consistent, χ' is obtained from χ by removing all occurrences of negative $\overline{p}$-literals.

$$\exists^+\overline{p}.(\chi \wedge \bigcirc(\Phi|\Psi)) = \chi' \wedge \bigcirc(\{\exists^+\overline{p}.\varphi \mid \varphi \in \Phi\}|\{\exists^+\overline{p}.\psi \mid \psi \in \Psi\})$$

when $(\chi \wedge \bigcirc(\Phi_1|\Psi)$ is weakly coherent, and $\bot$ when otherwise, with χ' is as before. Lastly

$$\exists^+\overline{p}.(\varphi \vee \psi) \quad = \quad \exists^+\overline{p}.\varphi \vee \exists^+\overline{p}.\psi$$

Lemma 6.4.29 *Every satisfiable syntactically closed $\mathcal{L}_r$-formula is semantically equivalent to a disjunction of complete and coherent DFDF-formulae.*

Proof. We may take the disjunction of all formulae $Char^{depth(\varphi)}_{\mathsf{PROP}(\varphi)}(\mathcal{M}, m)$ for which $(\mathcal{M}, m) \Vdash \varphi$. QED

Theorem 6.4.30 *If φ is a complete SDF-formula and $(\mathcal{M}, m)$ is any model, then*

$$(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$$

if and only if the exists a model $(\mathcal{N}, n)$ such that

$$(\mathcal{N}, n) \rightrightarrows^{depth(\varphi)}_{SC(\overline{p})(\mathit{PROP}(\varphi),\emptyset)} (\mathcal{M}, m) \text{ and } (\mathcal{N}, n) \Vdash \varphi.$$

Proof. We proceed by induction on φ. The base case, when φ is a conjunction of literals χ, is clear. (The cases for $\bot$ and $\top$ do not occur since φ is complete.) So suppose that φ is of the from $\chi \wedge \bigcirc(\Phi \mid \Psi)$, i.e. φ is a complete DFDF-formula. The subcase when φ is incoherent follows by theorem 6.4.27 and the definition of $\exists^+\overline{p}$. Hence, suppose that φ is coherent.

It is clear that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$ whenever $(\mathcal{N}, n) \Vdash \varphi$. Moreover, since $\exists^+\overline{p}.\varphi$ is syntactically closed and in positive in $\overline{p}$, it is preserved under syntactically closed $\overline{p}$-simulations. Hence the direction from bottom to top.

Conversely, suppose that $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$. By lemma 6.4.29, φ is a equivalent to a disjunction of complete and coherent formulae, say $\bigvee \varphi_i$. By the definition of $\exists^+\overline{p}$, it follows that $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi_i$, for some i.

Now $(\mathcal{M}, m) \Vdash Char^{depth(\varphi)}_{\mathsf{PROP}(\varphi)}(\mathcal{M}, m)$. Let us write κ for $Char^{depth(\varphi)}_{\mathsf{PROP}(\varphi)}(\mathcal{M}, m)$. By lemma 6.4.23 we have

$$(\mathcal{SM}(\kappa), w_\kappa) \rightrightarrows^{depth(\varphi)}_{SC(\emptyset)(\mathsf{PROP}(\varphi),\emptyset)} (\mathcal{M}, m).$$

Since φ_i is complete and coherent and $\exists^+\overline{p}.\varphi_i$ is obtained from it by deleting all negative $\overline{p}$-literals, we have $\varphi_i \preceq_{\overline{p}} \kappa$. Hence, by lemma 6.4.20,

$$(\mathcal{SM}(\varphi_i), w_{\varphi_i}) \rightrightarrows^{depth(\varphi_i)}_{SC(\overline{p})(\mathsf{PROP}(\varphi_i),\emptyset)} (\mathcal{SM}(\kappa), w_\kappa).$$

It follows that

$$(\mathcal{SM}(\varphi_i), w_{\varphi_i}) \rightrightarrows^{depth(\varphi_i)}_{SC(\overline{p})(\mathsf{PROP}(\varphi_i),\emptyset)} (\mathcal{M}, m).$$

Moreover, by lemma 6.4.23, $(\mathcal{SM}(\varphi_i), w_{\varphi_i}) \Vdash \varphi_i$, and hence $(\mathcal{SM}(\varphi_i), w_{\varphi_i}) \Vdash \varphi$. Hence we may take $(\mathcal{SM}(\varphi_i), w_{\varphi_i})$ as the desired model $(\mathcal{N}, n)$. This concludes the inductive step for φ of the form $\chi \wedge \bigcirc(\Phi \mid \Psi)$. The inductive step for disjunction is trivial. QED

The following corollary, which is the 'Lyndon-theorem' referred to in the title of this subsection, guarantees the existence of effectively obtainable positive equivalents for upward monotone syntactically closed $\mathcal{L}_r$-formulae.

Corollary 6.4.31 *If φ is a SDF-formula which is upward monotone in the propositional variables $\overline{p}$, then $\varphi \equiv_{\mathrm{sem}} \exists^+\overline{p}.\varphi$.*

Corollary 6.4.31, moreover, gives us a way to compute syntactically correct equivalents for propositionally downward monotone, syntactically open $\mathcal{L}^n_r$-formulae. The procedure for separately monotone formulae, as outlined in the previous section, is adapted in the obvious way — we will not bore the reader by reiterating the details. Here are a few examples:

Example 6.4.32 Consider the syntactically closed formula $\neg p \wedge \Diamond^{-1}\Box p$. This formula is unsatisfiable and hence upward monotone in p. Let us compute an equivalent for it using simulation quantifiers. First rewrite the formula equivalently as a complete SDF-formula:

$$\begin{aligned}
& \neg p \wedge \Diamond^{-1}\Box p \\
\equiv\ & (\neg p \wedge \bigcirc(\emptyset \mid \Box p)) \vee (\neg p \wedge \bigcirc(\top \mid \Box p)) \\
\equiv\ & \bigvee_{\chi\in\Gamma} (\neg p \wedge \bigcirc(\emptyset \mid \chi \wedge \bigcirc(\emptyset \mid \emptyset))) \vee \bigvee_{\chi\in\Gamma} (\neg p \wedge \bigcirc(\emptyset \mid \chi \wedge \bigcirc(p \mid \emptyset))) \\
& \vee \bigvee_{\chi\in\Gamma} \bigvee_{\psi\in\Psi} (\neg p \wedge \bigcirc(\psi \mid \chi \wedge \bigcirc(\emptyset \mid \emptyset))) \vee \bigvee_{\chi\in\Gamma} \bigvee_{\psi\in\Psi} (\neg p \wedge \bigcirc(\psi \mid \chi \wedge \bigcirc(p \mid \emptyset)))
\end{aligned}$$

where $\Gamma = \{p, \neg p\}$ and Ψ contains all $\{p\}$-full complete DFDF-formulae of depth 1. Now each disjunct γ of this formula is incoherent. Indeed, $\mathsf{Past}(\gamma)$ is either $\chi \wedge \bigcirc(\emptyset \mid \emptyset)$ of $\chi \wedge \bigcirc(p \mid \emptyset)$. In the first case $\mathsf{Fut}(\chi \wedge \bigcirc(\emptyset \mid \emptyset)) = \emptyset$ and hence there is no formula $\delta \in \mathsf{Fut}(\chi \wedge \bigcirc(\emptyset \mid \emptyset))$ such that $\gamma \leq \delta$. In the second case $\mathsf{Fut}(\chi \wedge \bigcirc(p \mid \emptyset)) = \{p\}$, and since $\mathsf{Bool}(\gamma) = \neg p$ there is similarly no formula $\delta \in \mathsf{Fut}(\chi \wedge \bigcirc(\emptyset \mid \emptyset))$ such that $\gamma \leq \delta$.

Thus applying the simulation quantifier $\exists^+ p$ to this formula, will return a disjunction in which is disjunct is $\perp$. $\lhd$

Example 6.4.33 Consider again the formula

$$\varphi = (\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$$

form examples 6.1.7 and 6.3.12. In example 6.3.12 it was pointed out that this formula is not separately monotone in p. However, it is propositionally monotone, for the formula

$$\mathsf{PSep}(\varphi) = (\Box^{-1}\neg r \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg r \vee \neg p)$$

is downward monotone in p. Indeed, this formula is semantically equivalent to $\Diamond\Box^{-1}(\Box^{-1}\neg r \vee \neg p)$. Let us compute an equivalent for φ. We begin by negating $\mathsf{PSep}(\varphi)$ to obtain

$$(\Diamond^{-1}r \wedge p \wedge \Box\neg p) \vee \Box\Diamond^{-1}(\Diamond^{-1}r \wedge p).$$

The obtained formula, which is syntactically closed, is rewritten as a complete SDF-formula:

$$(p \wedge \bigcirc(\emptyset \mid r)) \vee (p \wedge \bigcirc(\neg p \mid r)) \vee (\Box\Diamond^{-1}(\Diamond^{-1}r \wedge p))^{\star}.$$

We will not bother to write out the translation and transformation into a complete SDF-formula of the last disjunct, as this disjunct is positive in p and hence unaffected by the application of the simulation quantifier $\exists^{+}p$. As for the first two disjuncts:

$$\begin{array}{rc} & \exists^{+}p.((p \wedge \bigcirc(\emptyset \mid r)) \vee (p \wedge \bigcirc(\neg p \mid r))) \\ = & (p \wedge \bigcirc(\emptyset \mid r)) \vee (p \wedge \bigcirc(\top \mid r)). \end{array}$$

Undoing the translation, we obtain

$$(p \wedge \Diamond^{-1}r) \vee \Box\Diamond^{-1}(\Diamond^{-1}r \wedge p).$$

Lastly, negation and the reversal of the PSep-function yields the formula

$$(\neg p \vee \Box^{-1}\neg\mathbf{i}) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p)$$

which is syntactically open, negative in p, and, as the reader can easily verify, semantically equivalent to φ. $\lhd$

6.5 Conclusion

In this chapter we have extended the SQEMA-algorithm with a stronger Ackermann-rule that does not require negativity in the propositional variable being eliminated, but only downward monotonicity. In order to be able to guarantee the d-persistence of the formulae on which the extended algorithm succeeds, we found it necessary to preserve the correct syntactic shape of the sequents obtained during execution. Hence, we needed to compute equivalents to the downward monotone sequents in the application of the semantic Ackermann-rule, which were negative in the propositional variable under consideration and syntactically open. To facilitate this we considered two stronger forms of monotonicity, viz. separated monotonicity and propositional monotonicity. Both separated and propositional monotonicity require $\mathcal{L}_r^n$-formulae to be monotone even if we 'forget' some of the special features of the semantics of this language. In the latter case we 'forget' that nominals are interpreted as singletons, and in the former also that $\Diamond$ and $\Diamond^{-1}$ are inverses. In each case we developed methods based on simulation quantifiers which could be used to compute the desired equivalents.

The question remains — can similar methods be found for $\mathcal{L}_r^n$ formulae which are only monotone in the ordinary sense? It seems possible that the method of disjunctive forms and simulation quantifiers can be extended. However, notions like coherency and the construction of the standards models become quite tricky. Specifically, we cannot merrily take disjoint unions to construct the desired models, since this may cause nominals not to be interpreted as singletons.

Lastly, the most important question which we left open in this chapter is of course whether the formulae on which SSQEMA (the algorithm without replacement) succeeds are d-persistent. It is not clear how this could be proved. On the other hand, SSQEMA succeeds on none of the well-known non-canonical but elementary formulae, hence a counter example will probably have to involve a new specimen of this rare type.

Conclusion

The forgoing chapters have been a study in what we chose to term 'algorithmic correspondence and completeness in modal logic'. For the most part, this was based on the SQEMA-algorithm and its various extensions and adaptations. This approach can be considered suitable and natural for the following reasons:

1. It employs (hybrid) modal languages for its computations. This has several advantages, namely:

   (a) By using languages which have only the minimal required expressive power, we minimize complexity.

   (b) In these languages, certain semantic properties, like monotonicity, are decidable, and can hence be effectively employed in computations. By contrast, monotonicity is only semi-decidable in first-order logic. (Note that, although the fragments of first-order logic into which the standard translations map modal and hybrid languages are well known to be decidable, manipulation of these translations during attempts to eliminate predicate variables may often lead out of these well behaved fragments, at least syntactically.)

   (c) Computations in these languages are, at least for those familiar with modal and hybrid logics, intuitive and user friendly — the modal structure of formulae and the accompanying modal intuitions are not lost by a direct translation into first-order logic, for example.

   (d) The use of these languages facilitates (topological) canonicity proofs.

2. It stays within the confines of the method of substitutions, and hence continues a well-established tradition within modal correspondence theory.

3. It subsumes and unifies most of the known syntactically specified classes of elementary and canonical modal and hybrid formulae, and also generates new such classes.

4. SQEMA and its adaptations to hybrid languages are based on relatively simple rules that transform formulae in an incremental fashion. As a result, it is quite easy to run these algorithms 'by hand', at least on relatively small formulae. Thus they provide convenient research tools — where one could previously say "Look, we can axiomatize this using Sahlqvist formulae!", one could now also say "Some SQEMA-formulae will be enough to axiomatize that!". More powerful versions, like the semantic extension, are

less practical for manual computations, and hence here computerized implementations become even more desirable.

We conclude with a list of open questions as well as some thoughts on what the way forward could hold:

Open questions

1. The procedure outlined in the proof of corollary 1.5.15 for determining whether a given modal formula is semantically equivalent to a Sahlqvist formula, has non-elementary runtime complexity. Does there exist a procedure for answering this question which has an elementary upper bound on its complexity?

2. Is it decidable whether a given modal formula is model / locally equivalent to a Sahlqvist formula?

3. As was seen in example 2.3.3, SQEMA may succeed on an input formula if variables are eliminated in one order, but may fail on the same formula if elimination is attempted in another order. For this reason the algorithm could backtrack and attempt all orders. In example 2.3.3 the auxiliary rules played an essential role. Is it still true that the order of elimination matters if we were to remove the auxiliary rules from the algorithm?

4. Are all modal formulae on which DLS/SCAN succeed canonical?

5. How can SQEMA be naturally extended to compute first-order equivalents relative to special frame classes, e.g. the transitive frames?

6. Are all formulae on which SSQEMA succeeds canonical?

The way forward

Separating elementarity and canonicity. There is in principle no reason why we should only consider algorithms that guarantee both the elementarity and canonicity of the formulae which they reduce. Since no inclusion holds between the classes of elementary and canonical formulae, attempts at decidable approximations of these classes will probably have to diverge at some stage.

Indeed, as we saw, there exist non-canonical van Benthem-formulae, i.e., non-canonical formulae which are amenable to the method of substitutions. Also, as we have remarked, the basic SQEMA-algorithm can be used without adaptation to produce first-order equivalents of hybrid formulae (with nominals only) which axiomatize incomplete logics.

It is possible to detect a certain algorithmic flavour in Jónsson's algebraic proof ([Jón94]) of the canonicity of the Sahlqvist formulae. This proof depends on the transformation of the algebraic equations corresponding to these formulae into a certain normal form, from which their canonicity can then be deduced. Jónsson also adapts this approach to prove the canonicity of certain non-elementary formulae. This method can be extended to the inductive formulae with ease, and it seems plausible that an algorithm for proving the canonicity of (possibly non-elementary) modal formulae can be developed along these lines.

Beyond the van-Benthem formulae. The fact that SQEMA stays within the class of van Benthem-formulae was mentioned above as one of its advantages. However, since we know that there exist elementary formulae which fall outside this class, this fact simultaneously indicates a limitation. The non-inclusion of the elementary formulae in the van Benthem-formulae is witnessed by the conjunction of the McKinsey formula (a modal reduction principle) and the transitivity axiom. This formula is hence not reducible to a first-order frame equivalent by means of the method of substitutions. Thus, the modal reduction principles interpreted over transitive frames represent the most conspicuous class of elementary (and canonical!) modal formulae containing members on which all known algorithms stumble. Algorithms that are able to deal with this class will probably have to go beyond the familiar pattern of equivalence preserving transformations and substitutions.

Specialized second-order quantifier elimination. We saw that the elementarity problem for modal formulae can be reformulated as a second-order quantifier elimination problem, and can hence be attacked using algorithms like SCAN and DLS. While these algorithms are designed to handle arbitrary second-order formulae, the translations of modal formulae represent only a fragment of that langauge. Particularly, the standard translation embeds $\mathcal{L}$ into the so-called guarded fragment of first-order logic ([AvBN98]). Apart from having various desirable model theoretic properties, this fragment is also decidable! Hence certain semantic properties of formulae in this fragment, like monotonicity, for example, are also decidable. Thus second-order quantifier elimination algorithms specialized to this fragment could combine the positive computational properties of modal logics on the one hand, with the flexibility of first-order syntax on the other.

Semi-algorithms. SQEMA and DLS are guaranteed to always terminate, and hence define decidable classes of formulae. In the search for ever better approximations of the classes we are interested in, we will have to give up termination at some stage, and employ semi-algorithms. SCAN, the c-resolution stage of which might run forever, is an example of such a procedure. Indeed, as we saw (corollary 1.5.4) the class of formulae amenable to the method of substitutions, i.e. the van Benthem-formulae, although recursively enumerable, is not decidable. Hence, the development of more efficient semi-algorithms to identify (or approximate) the van Benthem-formulae might be a good place to start.

Richer languages. The languages treated in this thesis represent only a small segment of the wide spectrum of modal languages that appear in the literature. There are richer hybrid languages with, e.g., binders, languages with graded modalities (or counting quantifiers), languages with the difference operator, propositional dynamic logic, more expressive temporal languages, and the diverse family of languages of description logic, to name but a few. There is scope to development the algorithmic approach to correspondence and completeness in all these directions.

Bibliography

[ABM00] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[Ack35] W. Ackermann. Untersuchung über das Eliminationsproblem der mathematischen Logic. *Mathematische Annalen*, 110:390–413, 1935.

[AvBN98] H. Andreka, J. van Benthem, and I. Nemeti. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[BS85] G. Boolos and G. Sambin. An incomplete system of modal logic. *Journal of Philosophical Logic*, 14:351–358, 1985.

[BT99] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 24(1):27–54, 1999.

[BvBW06] P. Blackburn, J.F.A.K. van Benthem, and F. Wolter. *Handbook of Modal Logic*. Elsevier, 2006.

[CC06] A. Chagrov and L. A. Chagrova. The truth about algorithmic problems in correspondence theory. In G. Governatori, I. Hodkinson, and Y. Venema, editors, *Advances in Modal Logic*, volume 6, pages 121–138. College Publications, 2006.

[CGV05] W. Conradie, V. Goranko, and D. Vakarelov. Elementary canonical formulae: a survey on syntactic, algorithmic, and model-theoretic aspects. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic*, volume 5, pages 17–51. Kings College, 2005.

[CGV06a] W. Conradie, V. Goranko, and D. Vakarelov. Algorithmic correspondence and completeness in modal logic I: The core algorithm SQEMA. *Logical Methods in Computer Science*, 2(1:5), 2006.

[CGV06b] W. Conradie, V. Goranko, and D. Vakarelov. Algorithmic correspondence and completeness in modal logic II. Polyadic and hybrid extensions of the algorithm SQEMA. *Journal of Logic and Computation*, 16:579–612, 2006.

[Cha91] L. A. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, 56:1261–1272, 1991.

[CK90] C.C. Chang and H. J. Keisler. *Model Theory*. North-Holland, 1990.

[Con06] W. Conradie. On the strength and scope of DLS. *Journal of Applied Non-Classical Logics*, 16(3-4):279–296, 2006.

[CZ93] A. Chagrov and M. Zakharyaschev. The undecidability of the disjunction property of propositional logics and other related problems. *Journal of Symbolic Logic*, 58:967–1002, 1993.

[CZ95] A. Chagrov and M. Zakharyaschev. Sahlqvist formulas are not so elementary even above S4. In L. Csirmaz, D. Gabbay, and M. de Rijke, editors, *Logic Colloquium '92*, pages 61–73. CSLI Publications, 1995.

[CZ97] A. Chagrov and M. Zakharyaschev. *Modal Logic*. Oxford, 1997.

[DL02] G. D'Agostino and G. Lenzi. On modal μ-calculus with explicit interpolants. Technical Report PP-2002-17, Institute for Logic, Language and Computaion, University of Amsterdam, 2002.

[DŁS97] P. Doherty, W. Łukaszewicz, and A. Szalas. Computing circumscription revisited: A reduction algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.

[dR93] M. de Rijke. *Diamonds and Defaults*. Kluwer Academic Publishers, 1993.

[dR97] M. de Rijke. *Advances in Intensional Logic*. Springer, 1997.

[dRK97] M. de Rijke and N. Kurtonina. Simulating without negation. *Journal of Logic and Computation*, 7:1–22, 1997.

[dRV95] M. de Rijke and Y. Venema. Sahlqvist's theorem for Boolean algebras with operators with an application to cylindric algebras. *Studia Logica*, 54:61–78, 1995.

[Eng96] T. Engel. Quantifier elimination in second-order predicate logic. Master's thesis, Max-Planck-Institut für Informatik, Saarbrüken, 1996.

[Esa74] L. L. Esakia. Topological Kripke models. *Soviet Mathematics Doklady*, 15(1):147–151, 1974.

[Fin75a] K. Fine. Normal forms in modal logic. *Notre Dame Journal of Formal Logic*, 16(2):229–237, 1975.

[Fin75b] K. Fine. Some connections between elementary and modal logic. In S Kanger, editor, *Proc. of the 3rd Scandinavian Logic Symposium, Uppsala 1973*, pages 110–143, 1975.

[Fin85] K. Fine. Logics containing **K**, part II. *Journal of Symbolic Logic*, 50(2):619–651, 1985.

[Geo06] D. Georgiev. An implementation of the algorithm SQEMA for computing first-order correspondences of modal formulas. Master's thesis, Sofia University, Faculty of mathematics and computer science, 2006.

[GG84] D. M. Gabbay and F. Guenther, editors. *Handbook of Philosophical Logic*. Reidel, 1984.

[GG93] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22:607–636, 1993.

[Ghi95] S. Ghilardi. An algebraic theory of normal forms. *Annals Pure and Applied Logic*, 71(3):189–245, 1995.

[GHSV04] V. Goranko, U. Hustadt, R. A. Schmidt, and D. Vakarelov. SCAN is complete for all Sahlqvist formulae. In R. Berghammer, B. Möller, and G. Struth, editors, *Revised Selected Papers of the 7th International Seminar on Relational Methods in Computer Science and the 2nd International Workshop on Applications of Kleene Algebra, Bad Malente, Germany, May 12-17, 2003*, pages 149–162, 2004.

[GHV03] R. Goldblatt, I. Hodkinson, and Y. Venema. On canonical modal logics that are not elementarily determined. *Logique et Analyse*, 181:77–101, 2003.

[GO92] D. M. Gabbay and H.-J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992.

[GO06] V. Goranko and M. Otto. *Model theory of modal logic*. 2006. In [BvBW06].

[Gol75] R. Goldblatt. First-order definability in modal logic. *Journal of Symbolic Logic*, 40:35–40, 1975.

[Gol91] R. Goldblatt. The McKinsey axiom is not canonical. *Journal of Symbolic Logic*, 56(2):554–562, 1991.

[GP92] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2(1):5–30, 1992.

[GPT87] G. Gargov, S. Passy, and T. Tinchev. Modal environment for Boolean speculations. In D. Skordev, editor, *Mathematical Logic and its applications*, pages 253–263. Plenum Press, 1987.

[GSS06] D. M. Gabbay, R. Schmidt, and A. Szalas. *Second-Order Quantifier Elimination: Mathematical Foundations, Computational Aspects and Applications*. 2006. Monograph Manuscript, To appear.

[Gus96] J. Gustafsson. Quantifier elimination in second-order predicate logic. Technical Report LiTH-MAT-R-96-04, Department of Mathematics, Linköping University, Sweden, 1996.

[GV01] V. Goranko and D. Vakarelov. Sahlqvist formulae in hybrid polyadic modal languages. *Journal of Logic and Computation*, 11(5):737–254, 2001.

[GV02] V. Goranko and D. Vakarelov. Sahlqvist formulas unleashed in polyadic modal languages. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 3, pages 221–240, Singapore, 2002. World Scientific.

[GV06] V. Goranko and D. Vakarelov. Elementary canonical formulae: Extending Sahlqvist theorem. *Annals of Pure and Applied Logic*, 141(1-2):180–217, 2006.

[HM92] J. Y. Halpern and Y. O. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[Hod93] W. Hodges. *Model Theory.* Cambridge University Press, 1993.

[Hod07] I. Hodkinson. Hybrid formulas and elementary generated modal logics. *To appear in the Notre Dame Journal of Formal Logic*, 2007.

[Jón94] B. Jónsson. On the canonicity of Sahlqvist identities. *Studia Logica*, 53:473–491, 1994.

[JW95] D. Janin and I. Walukiewicz. Automata for the modal μ-calculus and related results. In *Mathematical foundations of computer science 1995 (Prague)*, volume 969 of *Lecture Notes in Computer Science*, pages 552–562. Springer, Berlin, 1995.

[Kra99] M. Kracht. *Tools and Techniques in Modal Logic.* Elsevier, 1999.

[Lem77] E.J. Lemmon. *An introduction to modal logic.* Blackwell, 1977.

[Lyn59a] R. C. Lyndon. An interpolation theorem in the predicate calculus. *Pacific Journal of Mathematics*, 9:129–142, 1959.

[Lyn59b] R. C. Lyndon. Properties preserved under homomorphism. *Pacific Journal of Mathematics*, 9:143–154, 1959.

[McC88] W. W. McCune. Un-skolemizing clause sets. *Information Processing Letters*, 29:257–263, 1988.

[Pit92] A. M. Pitts. On an interpretation of second order quantification in first order intuitionistic propositional logic. *Journal of Symbolic Logic*, 57(1):33–52, 1992.

[Sah75] H. Sahlqvist. Correspondence and completeness in the first and second-order semantics for modal logic. In S. Kanger, editor, *Proceedings of the 3rd Scandinavian Logic Symposium, Uppsala 1973*, pages 110–143, Amsterdam, 1975. Springer-Verlag.

[Spa93] E. Spaan. *The complexity of propositional tense logics.* 1993. In [dR93].

[SV89] G. Sambin and V. Vaccaro. A new proof of Sahlqvist's theorem on modal definability and completeness. *Journal of Symbolic Logic*, 54:992–999, 1989.

[Sza93] A. Szalas. On the correspondence between modal and classical logic: An automated approach. *Journal of Logic and Computation*, 3:605–620, 1993.

[Sza02] A. Szalas. On the correspondence between modal and classical logic: An automated approach. In S. Flesca and G. Ianni, editors, *Proceedings of JELIA'02*, pages 223–232. Springer-Verlag, 2002.

[tC05a] B. ten Cate. A note on the length of explicit definitions in modal logic. Unpublished note, 2005.

[tC05b] B. D. ten Cate. *Model Theory for Extended Modal Languages*. PhD thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2005.

[tCMV05] B. ten Cate, M. Marx, and P. Viana. Hybrid logics with Sahlqvist axioms. *Logic Journal of the IGPL*, 13(3):293–300, 2005.

[Tho75] S.K. Thomason. Reduction of second-order logic to modal logic. *Zeitschrift für mathematische Logic und Grundlagen der Mathematik*, 21:107–114, 1975.

[Vak03a] D. Vakarelov. Modal definability in languages with a finite number of propositional variables, and a new extension of the Sahlqvist class. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 4, pages 495–518. Kings College Publications, 2003.

[Vak03b] D. Vakarelov. On a generalization of the Ackermann's lemma for computing first-order equivalents of modal formulae. In *TARSKI Workshop, March 11 - 13, 2003, Toulouse, France*, 2003.

[Vak05] D. Vakarelov. Modal definability, solving equations in modal algebras and generalizations of the Ackermann lemma. In *Proceedings of the 5th Panhellenic Logic Symposium, July 25-28, 2005, Athens*, 2005.

[vB76] J. F. A. K. van Benthem. Modal reduction principles. *Journal of Symbolic Logic*, 41(2):301–312, 1976.

[vB83] J. F. A. K. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.

[vB84] J. F. A. K. van Benthem. *Correspondence Theory*. 1984. In [GG84].

[vB06] J. F. A. K. van Benthem. Modal frame correspondence and fixed-points. *Studia Logica*, 83:133–155, 2006.

[Vis96] A. Visser. Uniform interpolation and layered bisimulation. In *Gödel '96 (Brno, 1996)*, volume 6 of *Lecture Notes Logic*, pages 139–164. Springer, Berlin, 1996.

[Wal70] W. J. Walkoe. Finite partially-ordered quantification. *Journal of Symbolic Logic*, 35(4):535–555, 1970.

[Wil04] S. Willard. *General Topology.* Dover Publications, 2004.

[Zak92] M. Zakharyaschev. Canonical formulas for K4. Part I: Basic results. *Journal of Symbolic Logic*, 57:1377–1402, 1992.

[Zak96] M. Zakharyaschev. Canonical formulas for K4. Part II: Cofinal subframe logics. *Journal of Symbolic Logic*, 61:421–449, 1996.

[Zak97a] M. Zakharyaschev. Canonical formulas for K4. Part III: The finite model property. *Journal of Symbolic Logic*, 62:950–975, 1997.

[Zak97b] M. Zakharyaschev. *Canonical formulas for modal and superintuitionistic logics: a short outline*. 1997. In [dR97].

Index