Algorithmic Correspondence and Completeness in Modal Logic. II. Polyadic and Hybrid Extensions of the Algorithm SQEMA

WILLEM CONRADIE, *Department of Mathematics, University of Johannesburg and School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa.*
E-mail: wec@rau.ac.za

VALENTIN GORANKO, *School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa.*
E-mail: goranko@maths.wits.ac.za

DIMITER VAKARELOV, *Faculty of Mathematics and Computer Science, Sofia University, Sofia, Bulgaria.*
E-mail: dvak@fmi.uni-sofia.bg

Abstract

In Conradie, Goranko, and Vakarelov (2006, *Logical Methods in Computer Science*, 2) we introduced a new algorithm, SQEMA, for computing first-order equivalents and proving the canonicity of modal formulae of the basic modal language. Here we extend SQEMA, first to arbitrary and reversive polyadic modal languages, and then to hybrid polyadic languages too. We present the algorithm, illustrate it with some examples, and prove its correctness with respect to local equivalence of the input and output formulae, its completeness with respect to the polyadic inductive formulae introduced in Goranko and Vakarelov (2001, *J. Logic. Comput.*, 11, 737–754) and Goranko and Vakarelov (2006, *Ann. Pure. Appl. Logic*, 141, 180–217), and the d-persistence (with respect to descriptive frames) of the formulae on which the algorithm succeeds. These results readily expand to completeness with respect to hybrid inductive polyadic formulae and di-persistence (with respect to discrete frames) in hybrid reversive polyadic languages.

Keywords: Algorithm SQEMA, polyadic modal logic, hybrid logic, first-order correspondence, canonicity, completeness.

1 Introduction

The correspondence theory between modal logic and first-order logic, and the theory of Kripke completeness of modal logics (e.g. [2, 30]) are two of the main directions of technical development of modal logic since the introduction of Kripke semantics in the early 60s. The best known classical result on both correspondence and completeness is *Sahlqvist's theorem* [24], which identifies a class of formulae (subsequently named after Sahlqvist) that are first-order definable and valid in their respective canonical frames, being persistent with respect to all descriptive general frames (d-persistent). Recently, the class of Sahlqvist formulae was extended to the so-called *inductive formulae*, introduced in [16, 18] for arbitrary

Vol. 16 No. 5,
Published online 23 September 2006 doi:10.1093/logcom/exl026

polyadic modal languages, and in [17] for hybrid polyadic modal languages. On the other hand, as it follows from results of Chagrova in [3], the class of first-order definable modal formulae is undecidable, and hence, any attempt at syntactic characterization of that class can be an approximation at best.

In [5] (see also the survey [4]) we developed a stronger, algorithmic approach towards identifying first-order definable modal formulae, which are d-persistent as well. In particular, we introduced an algorithm called SQEMA (**S**econd-Order **Q**uantifier **E**limination for **M**odal formulae using **A**ckermann's lemma) for computing the first-order frame correspondents of modal formulae. We then proved the correctness of the algorithm and the d-persistence of the formulae on which it succeeds, and its completeness for the classes of monadic inductive (in particular, Sahlqvist) formulae.

With respect to the first-order correspondence, our approach was preceded and influenced by two earlier-developed algorithms for the elimination of second-order quantifiers over predicate variables, viz. SCAN [10, 12] and DLS [9, 14, 20, 21, 22, 23]. Each of them, applied to the negation of the standard translation of a modal formula into monadic second-order logic, attempts to eliminate all occurring existentially quantified predicate variables and thus to compute a first-order correspondent. To that aim, SCAN employs a modification of the resolution method, called constraint resolution, combined with a so-called 'purity deletion' rule which enables disposal of 'used-up' clauses, while DLS is based on a result by Ackermann [1] (see also the references above, as well as [5]), allowing explicit elimination, up to logical equivalence, of an existentially quantified second-order predicate variable. A modal version of Ackermann's lemma, that yielded the main transformation rule used by SQEMA, was proved in [5] (and in an algebraic context in [27, 28]; see also [23] for a somewhat different modal version).

An implementation of a variant of SQEMA (currently, for the ordinary modal language with nominals) has recently been announced in [13]. The program was realized by Dimiter Georgiev as a master project, and works online at http://fmi.uni-sofia.bg/fmi/logic/sqema.

In this article, we extend and modify SQEMA to arbitrary and reversive polyadic modal languages, with and without nominals. We prove the correctness of SQEMA with respect to local equivalence of the input and output formulae, and completeness with respect to the (hybrid) polyadic inductive formulae introduced in [17] and [18]. We then establish the d-persistence (with respect to descriptive frames) of the formulae on which the algorithm succeeds.

For hybrid (polyadic) modal languages, instead of d-persistence the useful property is di-persistence, i.e. persistence with respect to *discrete* general frames, because the special rules used for the axiomatization of the nominals cause the canonical general frame to be discrete, rather than descriptive. In the case of reversive polyadic languages the proof of di-persistence is unproblematic. For the general case, however, we have had to restrict the algorithm in order to guarantee di-persistence, which otherwise is generally not the case, even for some simple Sahlqvist formulae. Finally, we discuss the extension of SQEMA with special rules for the universal modality and the satisfaction operator.

2 Preliminaries

2.1 Syntax, semantics, and standard translations of polyadic and hybrid languages

A modal *similarity type* $\tau = (O, \rho_0)$ consists of a non-empty 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 $\bot$, a unary one ι_1, and a binary one ι_2. As it will become clear from the semantics below, the special modal term $\bot$ 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.

Definition 2.1
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* $\mathrm{MF}_\tau(\Theta)$ as follows:

(MT i) Every basic modal term from O is a modal term of the predefined arity.
(MT ii) Every formula containing no variables (variable-free formula) is a 0-ary modal term.
(MT iii) If $n > 0$, $\alpha(\beta_1, \ldots, \beta_n) \in MT_\tau$ 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*.

(MF i) Every propositional variable is a modal formula.
(MF ii) Every 0-ary modal term in MT_τ is a modal formula.
(MF iii) If φ is a formula, then $\neg\varphi$ is a formula.
(MF iv) If φ, ψ are formulae, then $\varphi \vee \psi$ is a formula.
(MF v) If $A_1, \ldots, A_n$ are formulae, α a modal term and $\rho(\alpha) = n > 0$, then $\langle\alpha\rangle(A_1, \ldots, A_n)$ is a modal formula.

The conjunction $\wedge$ is defined as usual. For any $\alpha \in \mathrm{MT}_\tau$, we will refer to $\langle\alpha\rangle$ as a *diamond (operator)*. The *dual* $[\alpha]$ of $\langle\alpha\rangle$, called a *box (operator)*, is defined by $[\alpha](\varphi_1, \ldots, \varphi_{\rho(\alpha)}) := \neg\langle\alpha\rangle(\neg\varphi_1, \ldots, \neg\varphi_{\rho(\alpha)})$.

The polyadic language so defined will be denoted by $\mathcal{L}_\tau(\Theta)$. If the particular set of proposition letters Θ over which the langauge is built is not important, we will omit it and simply write $\mathcal{L}_\tau$.

Note that variable-free formulae and 0-ary terms are regarded as both modal terms and formulae. This ambiguity of the syntax, admitted for the sake of technical simplicity and convenience, should not cause confusion if properly handled.

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 $n > 1$.

The *reversive extension* $\mathcal{L}_{\tau r}$ of the language $\mathcal{L}_\tau$ is defined by extending the definition of MT_τ with the clause:

(MT iv) 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.

The resulting set of modal terms will be denoted by $\mathrm{MT}_{\tau r}$. The diamond operator $\langle\alpha^{-j}\rangle$ is called the j-th inverse of $\langle\alpha\rangle$. Inverse boxes are defined as expected: $[\alpha^{-j}](\varphi_1, \ldots, \varphi_{\rho(\alpha)}) := \neg\langle\alpha^{-j}\rangle(\neg\varphi_1, \ldots, \neg\varphi_{\rho(\alpha)})$.

Note that in $\mathrm{MT}_{\tau r}$ we only require existence of inverses for modal terms from MT_τ. In general (unless all modal terms are unary), not every modal term in $\mathrm{MT}_{\tau r}$ has inverses there, even up to semantic equivalence, e.g. $(\alpha^{-j})^{-k}$, for $\alpha \in \mathrm{MT}_\tau$ and $j \neq k$, has no equivalent in $\mathrm{MT}_{\tau r}$.

Furthermore, we can allow full closure under inverses, by means of the modified clause.

(MT v) If α is a modal term of arity $n > 0$, then $\alpha^{-1}, \ldots, \alpha^{-n}$ are modal terms of arity n.

The result is the *completely reversive extension* $\mathcal{L}_{r(\tau)}$ of $\mathcal{L}_\tau$ with a set of modal terms $\mathrm{MT}_{r(\tau)}$. Such languages will be called *reversive (polyadic) languages*.

An occurrence of a modal operator or a subformula in a formula φ has *positive polarity* (or, is *positive*) if it is in the scope of an even number of negations; respectively, it has *negative polarity* (or, is *negative*) if it is in the scope of an odd number of negations.

Given a set of *nominals*, *Nom* (e.g. [2, 15]) disjoint with Θ, the *hybrid language* $\mathcal{L}^n_\tau(\Theta, Nom)$ extends $\mathcal{L}_\tau(\Theta)$ by adding the clause that every nominal is a formula. Hereafter, reference to both Θ and *Nom* will be suppressed whenever these are not essential or are clear from the context. Nominals will be denoted by boldface roman letters $\mathbf{i}, \mathbf{j}, \mathbf{k}, \ldots$, possibly indexed.

The reversive extension $\mathcal{L}^n_{\tau r}$ and the completely reversive extension $\mathcal{L}^n_{r(\tau)}$ of $\mathcal{L}^n_\tau$ are defined as above.

Besides nominals, hybrid languages usually involve the *universal modality* $[\cup]$ with semantics

$$(\mathcal{M}, w) \Vdash [\cup]\varphi \text{ iff } (\mathcal{M}, v) \Vdash \varphi \text{ for every } v \in \mathcal{M},$$

or the *satisfaction operator* @ with semantics

$$(\mathcal{M}, w) \Vdash @_c\varphi \text{ iff } (\mathcal{M}, v) \Vdash \varphi, \text{ where } v \text{ is the valuation of the nominal } c.$$

Hybrid languages with $[\cup]$ and @ will be revisited in Section 6.2.

Hereafter, we will refer to $\mathcal{L}_\tau, \mathcal{L}_{\tau r}, \mathcal{L}_{r(\tau)}, \mathcal{L}^n_\tau$, and $\mathcal{L}^n_{\tau r}$ as *the sublanguages* of $\mathcal{L}^n_{r(\tau)}$. When there is no essential difference for the various sublanguages, we will often formulate concepts and will give definitions with reference only to the full language $\mathcal{L}^n_{r(\tau)}$.

The set of propositional variables (respectively, nominals) occurring in a formula φ will be denoted by $\mathsf{PROP}(\varphi)$ (respectively, $\mathsf{NOM}(\varphi)$). A formula in $\mathcal{L}^n_{r(\tau)}$ is called *pure* if it contains no propositional variables, but (possibly) only nominals. Note that every pure formula is also a 0-ary modal term.

If $A, B(p) \in \mathcal{L}^n_{r(\tau)}$, we will write $B(A/p)$, or simply $B(A)$, for the formula obtained from $B(p)$ by uniform substitution of A for all occurrences of p.

Given a modal similarity type τ, a *(Kripke) τ-frame* is a structure $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}_\tau})$, consisting of a non-empty set W of *possible worlds* and, for each modal term $\alpha \in \mathrm{MT}_\tau$, 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 presented below (formally, by a simultaneous induction with this definition). Finally, the relation associated with a composite modal term is defined as follows:

$R_{\alpha(\beta_1,\ldots,\beta_n)} = \{(w, w_{11}, \ldots, w_{1b_1}, \ldots, w_{n1}, \ldots, w_{nb_n}) \subseteq W^{b_1+\cdots+b_n+1} \mid$
$\exists u_1 \ldots \exists u_n (R_\alpha w u_1 \ldots u_n \wedge \bigwedge_{i=1}^n R_{\beta_i} u_i w_{i1} \ldots w_{ib_i})\}$, where $\rho(\beta_i) = b_i$, $i = 1, \ldots, n$.

A *Kripke model based on a τ-frame* $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}_\tau})$ is a pair $\mathcal{M} = (\mathfrak{F}, V)$ (equivalently, a triple $(W, \{R_\alpha\}_{\alpha \in \mathrm{MT}_\tau}, V)$), 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. A *pointed τ-frame* $(\mathfrak{F}, w)$ is a pair consisting a frame $\mathfrak{F}$ together with a distinguished point $w \in W$. A pointed τ-model $(\mathcal{M}, w)$ is defined similarly.

The *truth of a formula* $\varphi \in \mathcal{L}_\tau$ in a pointed τ-model $(\mathcal{M}, w) = ((W, \{R_\alpha\}_{\alpha \in \mathrm{MT}_\tau}, V), w)$, denoted $(\mathcal{M}, w) \Vdash \varphi$, is defined recursively as follows:

- $(\mathcal{M}, w) \Vdash p$ iff $m \in V(p)$;
- $(\mathcal{M}, w) \nVdash \bot$;
- $(\mathcal{M}, w) \Vdash \neg \varphi$ iff $(\mathcal{M}, w) \nVdash \varphi$;
- $(\mathcal{M}, w) \Vdash \varphi \vee \psi$ iff $(\mathcal{M}, w) \Vdash \varphi$ or $(\mathcal{M}, w) \Vdash \psi$;
- $(\mathcal{M}, w) \Vdash \langle \alpha \rangle (\varphi_1, \ldots, \varphi_{\rho(\alpha)})$ if there exit $w_1, \ldots, w_{\rho(\alpha)}$ such that $R_\alpha(w, w_1, \ldots, w_{\rho(\alpha)})$ and $(\mathcal{M}, w_i) \Vdash \varphi_i$, for each $1 \leq i \leq \rho(\alpha)$.

Consequently, $(\mathcal{M}, w) \Vdash [\alpha](\varphi_1, \ldots, \varphi_{\rho(\alpha)})$ if, *for all* $w_1, \ldots, w_{\rho(\alpha)}$ such that $R_\alpha(w, w_1, \ldots, w_{\rho(\alpha)})$, it is the case that $(\mathcal{M}, w_i) \Vdash \varphi_i$, for *some* $1 \leq i \leq \rho(\alpha)$.

Note that, in terms of this semantics, the formulae $\langle \iota_1 \rangle (p)$ and $[\iota_1](p)$ are equivalent to p, and $\langle \iota_2 \rangle (p, q)$ is equivalent to $p \wedge q$, while $[\iota_2](p, q)$ is equivalent to $p \vee q$.

In the case of a reversive extension $\mathcal{L}_{\tau r}$, we specify that for any $\alpha \in \mathrm{MT}_\tau$ and $1 \leq j \leq \rho(\alpha)$,

$$R_\alpha^{-j}(w, v_1, \ldots, \ldots, v_{\rho(\alpha)}) \text{ holds iff } R_\alpha(v_j, v_1, \ldots, v_{j-1}, w, v_{j+1}, \ldots, v_{\rho(\alpha)}),$$

and extend the truth definition with the clause: $(\mathcal{M}, w) \Vdash \langle \alpha^{-j} \rangle (\varphi_1, \ldots, \varphi_{\rho(\alpha)})$ if there exist $w_1, \ldots, w_{\rho(\alpha)}$ such that $R_\alpha^{-j}(w, w_1, \ldots, w_{\rho(\alpha)})$ and $(\mathcal{M}, w_i) \Vdash \varphi_i$ for every i such that $1 \leq i \leq \rho(\alpha)$. The extension of the semantics to completely reversive extension $\mathcal{L}_{r(\tau)}$ is analogous.

To interpret languages with nominals, we extend the notion of model so that the valuations now assign subsets of the domain not only to propositional variables, but also to the nominals, with the restriction that every nominal must be assigned a *singleton*. Thus, nominals act as names for states. The truth definition is accordingly extended with the clause $(\mathcal{M}, w) \Vdash \mathbf{j}$ iff $V(\mathbf{j}) = \{w\}$.

A formula φ in (any of the sublanguages of) $\mathcal{L}^n_{r(\tau)}$ 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 $(\mathfrak{F}, V) \Vdash \varphi$.

Let $(\mathfrak{F}, w)$ be a pointed frame with domain W, and $X_1, \ldots, X_n, \subseteq W$ and $w_1, \ldots, w_m \in W$. Let v be a partial valuation in $\mathfrak{F}$ assigning $v(p_1) := X_1, \ldots, v(p_n) := X_n, v(\mathbf{i}_1) := w_1, v(\mathbf{i}_m) := w_m$. We say that an $\mathcal{L}^n_{\tau r}$-formula φ is *v-satisfiable on* $(\mathfrak{F}, w)$ if there exists a valuation V on $(\mathfrak{F}, w)$ extending v and such that $((\mathfrak{F}, V), w) \Vdash \varphi$. Sometimes we will write explicitly '$[p_1 := X_1, \ldots, p_n := X_n, \mathbf{i}_1 := w_1, \mathbf{i}_m := w_m]$-satisfiable'. Satisfiability with fixed parameters like this will be referred to as *parameterized satisfiability*. *Global parameterized satisfiability* as well as local and global *parameterized validity* are defined similarly.

To facilitate some proofs in the study, 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 consider 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 $\overline{\sigma}$. Without further ado, we will identify the modal terms $\alpha^{\overline{\overline{\sigma}}}$ and α. 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

$R^{\sigma}_{\alpha}m, m_1, \ldots, m_n$ and $(\mathcal{M}, m_i) \Vdash \varphi_i$ for all $1 \leq i \leq n$. Since every permutation is obtainable by the repeated transpositions (swapping of positions) which, in turn, are definable by means of composing inverses, the language so obtained and $\mathcal{L}_{r(\tau)}$ are equally expressive.[1]

The next lemma generalizes the fact enshrined as axiom R1 in [18].

LEMMA 2.2
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 $\bar{\sigma}(0) = j \neq 0$. 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 $m_1, \ldots, m_n \in \mathcal{M}$ such that $R^{\sigma}_{\alpha}m_0, m_1, \ldots m_n$, 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 $(\mathcal{M}, m_k) \Vdash \neg C$. 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 $(\mathcal{M}, m_0) \Vdash A$. ⊣

For a $\varphi \in \mathcal{L}^n_{r(\tau)}$ and a τ-model $\mathcal{M}$ we write $[\![\varphi]\!]_{\mathcal{M}} = \{w \in \mathcal{M} : (\mathcal{M}, w) \Vdash \varphi\}$ for the *extension* (or *truth-set*) of φ in $\mathcal{M}$. A formula $\varphi \in \mathcal{L}^n_{r(\tau)}$ is said to be *upward (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.

A *general τ-frame* is a structure $\mathfrak{F} = (W, \{R_{\alpha}\}_{\alpha \in \mathrm{MT}_{\tau}}, \mathbb{W})$ where $(W, \{R_{\alpha}\}_{\alpha \in \mathrm{MT}_{\tau}})$ is a τ-frame, and $\mathbb{W}$ is a Boolean algebra of subsets of 2^W, called *the admissible sets* in $\mathfrak{F}$, also closed under the modal operators $\langle\alpha\rangle, \alpha \in MT_{\tau}$, defined as follows:

$$\langle\alpha\rangle(X_1, \ldots, X_{\rho(\alpha)}) = \{y \in W : R_{\alpha}(y, x_1, \ldots, x_{\rho(\alpha)}) \text{ for some } x_1 \in X_1, \ldots, x_{\rho(\alpha)} \in X_{\rho(\alpha)}\}.$$

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

$$[\alpha](X_1, \ldots, X_{\rho(\alpha)}) = \{y \in W : x_1 \in X_1 \text{ or } \cdots \text{ or } x_{\rho(\alpha)} \in X_{\rho(\alpha)} \text{ whenever } R_{\alpha}(y, x_1, \ldots, x_{\rho(\alpha)})\}.$$

General τr-frames and general $\tau(r)$-frames are defined analogously, and will also be called *reversively extended general τ-frames*, and *reversive general τ-frames*, respectively.

[1] In the concluding section we mention how operators, called *transposers*, effecting such transpositions of arguments, can be used explicitly to simplify some technicalities.

Thus, the algebra of admissible sets of a reversively extended general τ-frame is closed under all $\langle R_{\alpha}^{-j}\rangle$ for $\alpha \in \mathrm{MT}_{\tau}$ (equivalently, under all $\langle\alpha\rangle, \alpha \in \mathrm{MT}_{\tau r}$). Similarly, the algebra of admissible sets of a reversive general τ-frame is closed under all $\langle\alpha\rangle, \alpha \in \mathrm{MT}_{\tau(r)}$.

The *underlying Kripke frame* of a general τ-frame $\mathfrak{F} = (W, \{R_{\alpha}\}_{\alpha\in\mathrm{MT}_{\tau}}, \mathbb{W})$ is the frame $\mathfrak{F}_{\sharp} := (W, \{R_{\alpha}\}_{\alpha\in\mathrm{MT}_{\tau}})$. A *model over* $\mathfrak{F}$ is a model over $\mathfrak{F}_{\sharp}$ 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; all these are defined likewise for reversively extended and reversive general frames.

Following [30] we define L_0 to be the first-order language with $=$, a family of predicates $\{R_{\alpha}\}_{\alpha\in\mathrm{MT}_{\tau}}$[2] of the respective arities, 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 $\{p_0, p_1, \ldots\}$. $\mathcal{L}_{\tau}$-formulae are translated into L_1 by means of the following *standard translation* function $\mathrm{ST}(\cdot,\cdot)$ which takes as arguments an $\mathcal{L}_{\tau}$-formula together with a variable from VAR:

- $\mathrm{ST}(p_i, x) := P_i(x)$ for every $p_i \in \mathsf{PROP}$;
- $\mathrm{ST}(\bot, x) := x \neq x$;
- $\mathrm{ST}(\neg\varphi, x) := \neg\mathrm{ST}(\varphi, x)$;
- $\mathrm{ST}(\varphi \vee \psi, x) := \mathrm{ST}(\varphi, x) \vee \mathrm{ST}(\psi, x)$;
- $\mathrm{ST}(\langle\alpha\rangle(\varphi_1, \ldots, \varphi_{\rho(\alpha)}), x) := \exists z_1, \ldots, \exists z_{\rho(\alpha)}(R_{\alpha}(x, z_1, \ldots, z_{\rho(\alpha)}) \wedge \bigwedge_{i=1}^{\rho(\alpha)} \mathrm{ST}(\varphi_i, z_i))$, where $z_1, \ldots, z_{\rho(\alpha)}$ are the first $\rho(\alpha)$ variables in VAR not appearing in $\mathrm{ST}(\varphi_1, x), \ldots, \mathrm{ST}(\varphi_{\rho(\alpha)}, x)$.

We extend $\mathrm{ST}(\cdot,\cdot)$ to reversively extended languages by adding the clause:

$$\mathrm{ST}(\langle\alpha^{-j}\rangle(\varphi_1, \ldots, \varphi_{\rho(\alpha)}), x) := \exists z_1, \ldots, \exists z_{\rho(\alpha)}(R^{-j}(x, z_1, \ldots, z_{\rho(\alpha)}) \wedge \bigwedge_{i=1}^{\rho(\alpha)} \mathrm{ST}(\varphi_i, z_i)).$$

The extension to reversive languages is analogous.

When dealing with languages containing nominals, we extend the standard translation with the clause $\mathrm{ST}(\mathbf{j}, x) := (x = y_{\mathbf{j}})$, where for each nominal $\mathbf{j}, y_{\mathbf{j}}$ is a reserved variable associated with it.

Now, for every model $(\mathcal{M}, w)$ and $\varphi \in \mathcal{L}^n_{r(\tau)}$, we have $(\mathcal{M}, w) \Vdash \varphi$ iff $\mathcal{M} \models \mathrm{ST}(\varphi, x)[x := w]$, and also $\mathcal{M} \Vdash \varphi$ iff $\mathcal{M} \models \forall x\mathrm{ST}(\varphi, x)$, if the variables $y_{\mathbf{j}}$ corresponding to the nominals are assigned the interpretations of their corresponding nominals. Thus, on Kripke models the modal language $\mathcal{L}^n_{r(\tau)}$ is a fragment of L_1. Furthermore, for every pointed frame $(\mathfrak{F}, w)$ and $\varphi \in \mathcal{L}^n_{r(\tau)}$, we have that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathfrak{F} \models \forall\overline{P}\forall\overline{y}\mathrm{ST}(\varphi, x)[x := w]$, and $\mathfrak{F} \Vdash \varphi$ iff $\mathfrak{F} \models \forall\overline{P}\,\forall\overline{y}\forall x\,\mathrm{ST}(\varphi, x)$, where $\overline{P}$ and $\overline{y}$ are, respectively, the tuples of all unary predicate symbols and all variables $y_{\mathbf{j}}$ corresponding to nominals, occurring in $\mathrm{ST}(\varphi, x)$. Thus, $\mathcal{L}^n_{r(\tau)}$-formulae express universal monadic second-order conditions on frames.

An $\mathcal{L}^n_{r(\tau)}$-formula φ and a first-order formula $\alpha(y) \in L_0$ are *local frame-correspondents* if for every pointed frame $(\mathfrak{F}, w)$,

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

Likewise, φ and a sentence $\alpha \in L_0$ are *global frame-correspondents* if for every frame $\mathfrak{F}$,

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

[2] Without risk of confusion, we will use the same notation for relations and respective predicate symbols.

An $\mathcal{L}^n_{r(\tau)}$-formula φ is *locally first-order definable*, if it has a local frame correspondent $\alpha(x) \in L_0$; *(globally) first-order definable*, if it has a global frame correspondent $\alpha \in L_0$. Note that every pure formula γ is locally first-order definable by the formula $\forall \bar{y} \mathrm{ST}(\gamma, x)$, where $\bar{y}$ is the tuple of all variables $y_{\mathbf{j}}$ corresponding to nominals $\mathbf{j}$ occurring in γ.

Two $\mathcal{L}^n_{r(\tau)}$-formulae are: *semantically equivalent* if they are true at the same states in the same models; *locally frame-equivalent* if they are valid at the same states in the same Kripke frames; *frame equivalent* if they are valid on the same frames.

3 Extending SQEMA to polyadic languages

In this section, we introduce the extended algorithm SQEMA that works on all formulae from $\mathcal{L}^n_{r(\tau)}$, for an arbitrary modal similarity type τ. We will give examples of the execution of these extensions, will prove their correctness with respect to local first-order equivalence, and will also prove that the full extension for $\mathcal{L}^n_{r(\tau)}$ is complete with respect to the class of polyadic inductive formulae in reversive hybrid languages, introduced in [17].

3.1 *The algorithm* SQEMA

The algorithm transforms sets of formulae, which we will call *systems of* SQEMA-*equations*, because, if we think of a formula A in which all propositional variables are implicitly existentially quantified as an algebraic equation $A = 1$, the procedure somewhat resembles solving systems of linear equations by Gauss' elimination method.

Algorithm SQEMA(φ). This is the main body of the algorithm. It takes an $\mathcal{L}^n_{r(\tau)}$-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 Preprocess, form the *initial system* $\| \neg \mathbf{i} \vee \alpha_k$, where $\mathbf{i}$ is a fixed, reserved nominal, not allowed to occur in any input formula. Then call subroutine Transform($\| \neg \mathbf{i} \vee \alpha_k$).
- **2.2** If Transform($\| \neg \mathbf{i} \vee \alpha_k$) 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 Transform($\| \neg \mathbf{i} \vee \alpha_k$) 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 ($\| \neg \mathbf{i} \vee \alpha_k$).
3.2. Call Postprocess($\{\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n\}$).
3.3. The subroutine 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 '$\leftrightarrow$' (if admitted in the language), 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 $\langle\alpha\rangle(\gamma_1, \ldots, \varphi \vee \psi, \ldots, \gamma_n) \equiv (\langle\alpha\rangle(\gamma_1, \ldots, \varphi, \ldots, \gamma_n) \vee \langle\alpha\rangle(\gamma_1, \ldots, \psi, \ldots, \gamma_n))$ 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 of SQEMA-equations Sys, if possible, and to return a pure formula.

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 Eliminate(Sys, p).

Transform.3. If Eliminate(Sys, p) has returned FAIL for every variable p remaining in the Sys, return FAIL;

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

Transform.3.1. Call Transform(Sys$'$).

Transform.3.2. If Transform(Sys$'$) returns FAIL, return FAIL;

else, if (Sys$'$) returns a pure system, Sys$''$, return Sys$''$.

Eliminate(Sys, p). This procedure takes as an input, a system of SQEMA-equations together with a propositional variable. The goal is, by applying the SQEMA-transformation rules (listed below), to rewrite the system of equations, Sys, 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 equation is either negative in p, or of the form $\alpha \vee p$, with p not occurring in α, i.e. to 'extract' p and 'solve' for it. If this can be achieved, the Ackermann-rule is applied, eliminating the variable p.

If this succeeds, returns the transformed system Sys$'$ from which p has been eliminated; else, returns FAIL.

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

Postprocessing.1. For each Sys$_k \in \{$Sys$_1, \ldots,$ Sys$_n\}$, form the pure formula pure$_k$, by taking the conjunction of all equations in the pure system Sys$_k$.

Postprocessing.2. Form the formula pure(φ) by taking the disjunction of the formulae pure$_k$, obtained in step Postprocessing.1.

Postprocessing.3. Form the formula $\forall\overline{y}\exists x$ST($\neg$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 **i**) left free, since a local correspondent is being computed. Return this first-order formula.

Remark 3.1
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 do not discuss such ordering strategies in this article, but assume that the choice is made non-deterministically, while allowing backtracking, thus exploring every possible order of elimination until either an order that succeeds is found, or all orders have failed.

3.2 *The transformation rules of* SQEMA

The transformation rules used by the algorithm are listed below. Note that these are *rewriting rules*, i.e. the equation above the line is replaced in the system by the equations listed below the line.

I. Rules for the logical connectives:

$$(\wedge\text{-rule}) \quad \frac{C \vee (A \wedge B)}{C \vee A,\, C \vee B}$$

$$(\text{Left-shift} \vee\text{-rule}) \frac{C \vee (A \vee B)}{(C \vee A) \vee B} \qquad (\text{Right-shift} \vee\text{-rule}) \frac{(C \vee A) \vee B}{C \vee (A \vee B)}$$

$$(\Box\text{-rule}) \frac{A \vee [\gamma](B_1, \ldots, B_n)}{[\gamma^{-i}](B_1, \ldots, B_{i-1}, A, B_{i+1}, \ldots, B_n) \vee B_i}$$

$$(\text{Inverse } \Box\text{-rule}) \frac{A \vee [\gamma^{-i}](B_1, \ldots, B_n)}{[\gamma](B_1, \ldots, B_{i-1}, A, B_{i+1}, \ldots, B_n) \vee B_i}$$

$$(\Diamond\text{-rule}) \quad \frac{\neg \mathbf{j} \vee \langle\gamma\rangle(A_1, \ldots, A_n)}{\neg \mathbf{j} \vee \langle\gamma\rangle(\mathbf{k}_1, \ldots, \mathbf{k}_n),\, \neg \mathbf{k}_1 \vee A_1, \ldots, \neg \mathbf{k}_n \vee A_n}$$

where $\mathbf{k}_1, \ldots, \mathbf{k}_n$ are nominals, not occurring in the premise. Note that for monadic modalities the $\Box$- and $\Diamond$-rules simplify as follows:

$$(\Box\text{-rule}) \quad \frac{A \vee [\alpha]B}{[\alpha^{-1}]A \vee B} \qquad (\text{Inverse } \Box\text{-rule}) \frac{A \vee [\alpha^{-1}]B}{[\alpha]A \vee B}$$

$$(\text{Monadic } \Diamond\text{-rule}) \quad \frac{\neg \mathbf{j} \vee \langle\alpha\rangle A}{\neg \mathbf{j} \vee \langle\alpha\rangle \mathbf{k},\, \neg \mathbf{k} \vee A}$$

where α is any unary modal term, and $\mathbf{k}$ is a nominal not occurring in the premise.

II. Ackermann-rule: This rule is based on the equivalence given in Ackermann's lemma. It works not on a single equation, but by transforming the part of the system consisting of all equations containing p, as follows:

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

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.

III. Polarity-switching-rule: Switch the polarity of every occurrence of a chosen variable p within the current system, i.e. replace $\neg p$ by p and p by $\neg p$ for every occurrence of p not prefixed by $\neg$.

IV. Auxiliary rules: These rules are intended to provide the algorithm with some propositional reasoning capabilities and to effect the duality between the modal operators; they are applied whenever enabled.

(1) Commutativity and associativity of $\wedge$ and $\vee$ (tacitly used).
(2) Replace $\gamma \vee \neg\gamma$ with $\top$ and $\gamma \wedge \neg\gamma$ with $\bot$.
(3) Replace $\gamma \vee \top$ with $\top$ and $\gamma \vee \bot$ with γ.
(4) Replace $\gamma \wedge \top$ with γ and $\gamma \wedge \bot$ with $\bot$.

Remark 3.2. We note that:

(1) Apart from the polarity-switching-rule, no transformation rule changes the polarity of any occurrence of a propositional variable.
(2) The application of any transformation rule to an equation in negation normal form yields again an equation in this normal form.
(3) By the previous comment and the fact that all equations in the initial systems are in negation normal form, if follows that in the application of the Ackermann-rule, the equations $B_1(p), \ldots, B_m(p)$ are still in negation normal form. Hence, the substitution $B_i((A_1 \wedge \ldots \wedge A_n)/\neg p)$, prescribed by the rule, will indeed eliminate *all* occurrences of the variable p from the system.

3.3 Examples

Here are two examples of executions of SQEMA:

Example 3.3. Consider the formula

$$\varphi_1 = [\mathbf{3}](\neg[\mathbf{1}]p, \neg[\mathbf{2}](\neg p, q), \langle\mathbf{1}\rangle[\mathbf{1}]q),$$

where **1**, **2**, **3** are modal terms of arities, respectively, 1, 2 and 3. This is an inductive formula (see further), which is not equivalent to a polyadic Sahlqvist formula in terms of [8]. Here is a successful execution of SQEMA on φ_1:

Step 1 Negating, and driving the negation inwards, we obtain

$$\neg\varphi_1 \equiv \langle\mathbf{3}\rangle([\mathbf{1}]p, [\mathbf{2}](\neg p, q), [\mathbf{1}]\langle\mathbf{1}\rangle\neg q).$$

This completes the Preprocessing phase.

Step 2 There is only one initial system of SQEMA-equations:

$$\|\neg\mathbf{i} \vee \langle\mathbf{3}\rangle([\mathbf{1}]p, [\mathbf{2}](\neg p, q), [\mathbf{1}]\langle\mathbf{1}\rangle\neg q).$$

Step 3 Applying the $\Diamond$-rule yields:

$$\left\| \begin{array}{l} \neg\mathbf{i} \vee \langle\mathbf{3}\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ \neg\mathbf{j}_1 \vee [\mathbf{1}]p \\ \neg\mathbf{j}_2 \vee [\mathbf{2}](\neg p, q) \\ \neg\mathbf{j}_3 \vee [\mathbf{1}]\langle\mathbf{1}\rangle\neg q \end{array} \right.$$

Step 4 We choose to eliminate p first, and apply the $\Box$-rule to the second equation:

$$\left\| \begin{array}{l} \neg\mathbf{i} \vee \langle\mathbf{3}\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ [\mathbf{1}^{-1}]\neg\mathbf{j}_1 \vee p \\ \neg\mathbf{j}_2 \vee [\mathbf{2}](\neg p, q) \\ \neg\mathbf{j}_3 \vee [\mathbf{1}]\langle\mathbf{1}\rangle\neg q \end{array} \right.$$

Step 5 The system is now ready for the application of the Ackermann-rule to the second and third equations, to eliminate p:

$$\left\| \begin{array}{l} \neg\mathbf{i} \vee \langle\mathbf{3}\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ \neg\mathbf{j}_2 \vee [\mathbf{2}]([\mathbf{1}^{-1}]\neg\mathbf{j}_1, q). \\ \neg\mathbf{j}_3 \vee [\mathbf{1}]\langle\mathbf{1}\rangle\neg q \end{array} \right.$$

Step 6 Now, we want to eliminate q too. To that aim, we transform the only equation where q is positive, by applying again the $\Box$-rule:

$$\left\| \begin{array}{l} \neg\mathbf{i} \vee \langle\mathbf{3}\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ [\mathbf{2}^{-2}]([\mathbf{1}^{-1}]\neg\mathbf{j}_1, \neg\mathbf{j}_2) \vee q. \\ \neg\mathbf{j}_3 \vee [\mathbf{1}]\langle\mathbf{1}\rangle\neg q \end{array} \right.$$

Step 7 Applying Ackermann-rule again eliminates q:

$$\left\| \begin{array}{l} \neg\mathbf{i} \vee \langle\mathbf{3}\rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3) \\ \neg\mathbf{j}_3 \vee [\mathbf{1}]\langle\mathbf{1}\rangle[\mathbf{2}^{-2}]([\mathbf{1}^{-1}]\neg\mathbf{j}_1, \neg\mathbf{j}_2). \end{array} \right.$$

Step 8 We thus obtain:

$$\mathsf{pure}(\varphi_1) = (\neg \mathbf{i} \vee \langle \mathbf{3} \rangle(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)) \wedge (\neg \mathbf{j}_3 \vee [\mathbf{1}]\langle \mathbf{1} \rangle [\mathbf{2}^{-2}]([\mathbf{1}^{-1}]\neg \mathbf{j}_1, \neg \mathbf{j}_2)).$$

Step 9 Negating and translating into first-order logic, we obtain a local first-order equivalent:

$$FO(\varphi_1)(x) = \forall y_1 y_2 y_3 (R_{\mathbf{3}} x y_1 y_2 y_3 \rightarrow \exists v (R_{\mathbf{1}} y_3 v \wedge \forall w (R_{\mathbf{1}} v w \rightarrow \exists s (R_{\mathbf{2}} y_2 s w \wedge R_{\mathbf{1}} y_1 s)))).$$

Example 3.4

Here is a formula on which SQEMA does not succeed, despite it being locally first-order definable.[3]

$$\varphi_2 = [\mathbf{2}](\neg [\mathbf{1}](\neg [\mathbf{1}]p \vee p), p \wedge [\mathbf{1}]\bot).$$

Here is an attempt to execute SQEMA on it:

Step 1 Negating, and driving the negation inwards, we obtain

$$\neg \varphi_2 \equiv \langle \mathbf{2} \rangle ([\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p), \neg p \vee \langle \mathbf{1} \rangle \top).$$

Step 2 The only initial system is:

$$\| \neg \mathbf{i} \vee \langle \mathbf{2} \rangle ([\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p), \neg p \vee \langle \mathbf{1} \rangle \top).$$

Step 3 Applying the $\Diamond$-rule yields:

$$\left\| \begin{array}{l} \neg \mathbf{i} \vee \langle \mathbf{2} \rangle (\mathbf{j}_1, \mathbf{j}_2) \\ \neg \mathbf{j}_1 \vee [\mathbf{1}](\langle \mathbf{1} \rangle \neg p \vee p) \\ \neg \mathbf{j}_2 \vee \neg p \vee \langle \mathbf{1} \rangle \top \end{array} \right. .$$

Step 4 We now apply the $\Box$-rule to the second equation:

$$\left\| \begin{array}{l} \neg \mathbf{i} \vee \langle \mathbf{2} \rangle (\mathbf{j}_1, \mathbf{j}_2) \\ [\mathbf{1}^{-1}]\neg \mathbf{j}_1 \vee \langle \mathbf{1} \rangle \neg p \vee p \\ \neg \mathbf{j}_2 \vee \neg p \vee \langle \mathbf{1} \rangle \top \end{array} \right. .$$

We are stuck now—the rules do not enable us to solve for p to prepare the system for the application of the Ackermann-rule in order to eliminate p.

4 Correctness and strength of SQEMA

4.1 Correctness

Here we justify the correctness of the algorithm in terms of local equivalence of the input modal formula to the returned first-order (L_0) formula.

---

[3] Actually, this formula is locally equivalent to the inductive formula (see next section) $[\mathbf{2}](\neg[\mathbf{1}]p, p \wedge [\mathbf{1}]\bot)$.

The following is a modal version of Ackermann's lemma [1] for the polyadic and hybrid languages, proved for the monadic case in [5] (see also [23]); the general proof is completely analogous.

Lemma 4.1 (Ackermann's lemma)
Let τ be a similarity type and A and B be formulae, both in (any sublanguage of) $\mathcal{L}^n_{r(\tau)}$, such that B is negative in p, and p does not occer in A. Then, for any Kripke model $\mathcal{M}$,

$$\mathcal{M} \models B(A/p)$$

if and only if there exists a model $\mathcal{M}'$, *differing from* $\mathcal{M}$ *at most in the valuation of* p, *such that*

$$\mathcal{M}' \models (\neg A \vee p) \wedge B(p).$$

Given a system of SQEMA-equations Sys, we write Form(Sys) for the formula obtained by taking the conjunction of all the equations in Sys.

Lemma 4.2
Let Sys be a system of SQEMA-equations, Sys′ be a system obtained from Sys by the application of a SQEMA-transformation rule, and let $(\mathfrak{F}, w)$ be a pointed Kripke frame. Then Form(Sys) is globally $[\mathbf{i} := w]$-satisfiable on $(\mathfrak{F}, w)$, if and only if Form(Sys′) is globally $[\mathbf{i} := w]$-satisfiable on $(\mathfrak{F}, w)$.

Proof. The proof is a routine verification that all transformation rules maintain this type of parameterized satisfiability on Krike frames. The case for the Ackermann-rule is justified by Lemma 4.1. ⊣

Theorem 4.3 (Correctness of SQEMA w.r.t. local equivalence)
If SQEMA succeeds on an input formula, φ, then φ is locally frame-correspondent to the returned first-order formula.

Proof. It suffices to prove the claim assuming, for simplicity, that φ does not produce disjunctive branching in the execution of SQEMA; the general case follows immediately. 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 *translation* $\mathrm{TR}(\mathsf{Sys}_j)$ of a system Sys_j to be the second-order formula $\exists \overline{P} \exists \overline{y} \forall x\, \mathrm{ST}(\mathsf{Form}(\mathsf{Sys}_j), x)$, where $\overline{P}$ is the tuple of all predicate variables and $\overline{y}$ the tuple of all variables corresponding to nominals *other than the reserved nominal* $\mathbf{i}$, occurring in $\mathsf{Form}(\mathsf{Sys}_j)$. 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 $\exists\overline{y}\forall x \mathrm{ST}(\mathsf{pure}(\varphi), x)$, where $\mathsf{pure}(\varphi)$ is the pure formula $\mathsf{Form}(\mathsf{Sys}_r)$. Now, for any pointed Kripke frame $(\mathfrak{F}, w)$ we have:

$(\mathfrak{F}, w) \Vdash \varphi$ iff
$\mathfrak{F} \models \forall\overline{P}\forall\overline{y}\mathrm{ST}(\varphi, x)[x := w]$ iff
$\mathfrak{F} \models \forall\overline{P}\forall\overline{y}\exists x\mathrm{ST}(\mathbf{i} \wedge \varphi, x)[y_\mathbf{i} := w]$ iff
$\mathfrak{F} \not\models \exists\overline{P}\exists\overline{y}\forall x\mathrm{ST}(\neg\mathbf{i} \vee \neg\varphi, x)[y_\mathbf{i} := w]$ iff
$\mathfrak{F} \not\models \mathrm{TR}(\mathsf{Sys}_1)[y_\mathbf{i} := w]$.

Now, rephrased in second-order logic, Lemma 4.2 says that:

$\mathfrak{F} \not\models \mathrm{TR}(\mathsf{Sys}_j)[y_\mathbf{i} := w]$ if and only if $\mathfrak{F} \not\models \mathrm{TR}(\mathsf{Sys}_{j+1})[y_\mathbf{i} := w]$, for all $1 \leq j < r$.

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

Thus, the formula $\forall\overline{y}\exists x\neg\mathrm{ST}(\mathsf{pure}(\varphi), x)$ which SQEMA returns is a local first-order correspondent for the input formula φ. Accordingly, $\forall y_{\mathbf{i}}\forall\overline{y}\exists x\neg\mathrm{ST}(\mathsf{pure}(\varphi), x)$ is a global first-order correspondent of φ.

4.2 *Completeness of* SQEMA *for hybrid polyadic inductive formulae*

In this section we show that SQEMA succeeds on every polyadic inductive formula, introduced in [17, 18], where it is also proved that such formulae are locally first-order definable and d-persistent.

Let us first briefly recall the definition of (hybrid) polyadic inductive formulae in $\mathcal{L}^n_{r(\tau)}$; the definition projects accordingly to all sublanguages.

First, we need some preliminary notions. A formula in $\mathcal{L}^n_{r(\tau)}$ is *variable-negative* if all variables in it have only negative occurrences, while nominals can have any occurrences. A formula $[\beta](N_1, \ldots, N_m)$, where β is an m-ary modal term and $N_1, \ldots, N_m$ are variable-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 β is an $(m+1)$-ary modal term, p is a propositional variable, and $N_1, \ldots, N_m$ are variable-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 4.4
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 4.5
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 4.6
An *inductive formula* is any regular formula with an acyclic dependency digraph.

Definition 4.7
Call a system of SQEMA-equations an *inductive system*, if it has the form

$$\left\|\begin{array}{l}\neg\mathbf{i}_1 \vee [\beta_1](p_1, N_{1_1}, \ldots, N_{1_m})\\ \vdots\\ \neg\mathbf{i}_n \vee [\beta_n](p_n, N_{n_1}, \ldots, N_{n_m})\\ \neg\mathbf{j}_1 \vee Neg_1\\ \vdots\\ \neg\mathbf{j}_k \vee Neg_k\end{array}\right.,$$

where either n or k, but obviously not both, may possibly be 0, each $[\beta_i](p_i, N_{i_1}, \dots, N_{i_m})$ is a headed box with head p_i such that the dependency digraph of this set of boxes is acyclic, every propositional variable occurring in the system occurs at least once as the head of some $[\beta_i](p_i, N_{i_1}, \dots, N_{i_m})$, and each Neg_i is a variable-negative formula.

For technical convenience, we assume that in the system above all heads of the boxes occur as first component of the boxes, which can always be arranged using transposers.

Lemma 4.8
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 equations of the form $\neg\mathbf{i}_i \vee [\beta_i](p_i, N_{i_1}, \dots, N_{i_m})$ 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 equation $\neg\mathbf{i}_i \vee [\beta_i](p_i, N_{i_1}, \dots, N_{i_m})$ that has $p_i = q$, replacing it in the system with $[\beta_i^{-1}](\neg\mathbf{i}_i, N_{i_1}, \dots, N_{i_m}) \vee p_i$, where $[\beta_i^{-1}](\neg\mathbf{i}_i, N_{i_1}, \dots, N_{i_m})$ is a pure formula, by the minimality of $q = p_i$. The Ackermann-rule now becomes applicable to these equations, i.e. we may remove them from the system and substitute $\bigwedge\{[\beta_i^{-1}](\neg\mathbf{i}_i, N_{i_1}, \dots, N_{i_m}) \mid p_i = q\}$ for all remaining occurrences of $\neg q$. It is not difficult to see that the system obtained in this way is still an inductive system, now containing at most $n - 1$ equations of the form $\neg\mathbf{i}_j \vee [\beta_j](p_j, N_{j_1}, \dots, N_{j_m})$. Finally, we appeal to the inductive hypothesis to conclude that all remaining variables can be eliminated by the application of SQEMA-transformation rules. ⊣

Theorem 4.9
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 $\|\neg\mathbf{i} \vee \langle\alpha\rangle(B_1, \dots, B_n)$, which, after application of the $\Diamond$-rule, becomes an inductive system. Now, we appeal to Lemma 4.8. ⊣

Corollary 4.10
SQEMA succeeds on all polyadic Sahlqvist formulae, as defined in [2].

Proof. As shown in [18], every polyadic Sahlqvist formulae, as defined in [2], is semantically equivalent to a conjunctions of polyadic inductive formulae. This equivalence is captured by SQEMA; we leave the details to the reader. ⊣

5 Languages without nominals and d-persistence

In this section, we prove that every input formula φ from $\mathcal{L}_\tau$ (respectively, $\mathcal{L}_{r(\tau)}$) on which SQEMA succeeds is *locally d-persistent*, i.e. locally persistent with respect to the class of descriptive frames (respectively, the class of reversive descriptive frames).[4] Recall (see e.g. [2])

[4] We will not consider the intermediate case of formulae from $\mathcal{L}_{\tau r}$, as we have introduced these languages only for technical purposes.

that d-persistence of a formula, usually also referred to as 'canonicity', is an important property, because every normal modal logic axiomatized with d-persistent formulae is strongly complete with respect to validity in its Kripke frames.

The proof generally follows the steps of the proof of d-persistence for the monadic case, presented in [5], but involves some technical overhead due to the polyadic modalities. Note that the property of d-persistence depends not only on the given formula, but also on the language in which it is considered, because the class of general, and in particular descriptive, frames depends on that language—the algebra of admissible sets is closed under the operators corresponding to the modal terms *of the language*. We will treat simultaneously arbitrary and reversive languages. While the latter case does not follow from the former, it is easier because there the inverse modalities, which come into play during the execution of SQEMA anyway, are already part of the input language, and therefore preserve admissibility of sets in reversive general frames.

5.1 The topology of descriptive frames

We fix a (polyadic) similarity type τ for the rest of this subsection. All concepts defined here apply likewise to arbitrary and to reversive languages; so when we do not wish to specify which is the case, we will denote the set of modal terms simply by MT.

With every general frame $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ 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 $C(\mathbb{W})$ denote the family of sets closed with respect to $T(\mathfrak{F})$.

A general frame $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ is *differentiated* if for every $x, y \in W$ such that $x \neq y$, there exists $X \in \mathbb{W}$ such that $x \in X$ and $y \notin X$; equivalently, if $T(\mathfrak{F})$ is a Hausdorff space.

Note that all singleton sets are closed in any differentiated frame.

A relation R_α in $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ 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 every $x \in W$,

$$R_\alpha x, x_1, \ldots, x_n \text{ iff } x \in \bigcap\{\langle\alpha\rangle(X_1, \ldots, X_n) \mid X_1, \ldots, X_n \in \mathbb{W} \ \& \ x_1 \in X_1, \ldots, x_n \in X_n\}.$$

Now, the general frame $\mathfrak{F}$ is *tight* if the relation R_α is tight in $\mathfrak{F}$ for every basic modal term α.

A general frame $\mathfrak{F}$ is *compact* if every family of admissible sets from $\mathbb{W}$ with the finite intersection property (FIP) has a non-empty intersection; equivalently, if $T(\mathfrak{F})$ is compact. $\mathfrak{F}$ is *descriptive* if it is differentiated, tight, and compact.

It has been proved in [18] that in any differentiated general frame, for any $\alpha \in \mathrm{MT}$ the relation R_α is tight iff for every $x \in W$ the set $R_\alpha(x) = \{(x_1, \ldots, x_n) \mid R_\alpha x x_1 \ldots x_n\}$ is closed, i.e. R_α is *point-closed*.

A formula φ is *locally d-persistent*, if, for every pointed descriptive frame $(\mathfrak{F}, w)$ for the respective language, it is the case that $(\mathfrak{F}_\sharp, w) \Vdash \varphi$ whenever $(\mathfrak{F}, w) \Vdash \varphi$; φ is *d-persistent* if $\mathfrak{F}_\sharp \Vdash \varphi$ whenever $\mathfrak{F} \Vdash \varphi$. Clearly, local d-persistence implies d-persistence.

Given any (general) frame $\mathfrak{F}$ with domain W, we can regard any $\mathcal{L}^n_{r(\tau)}$-formula $\varphi(p_1, \ldots, p_n, \mathbf{i}_1, \ldots, \mathbf{i}_m)$ as a set-theoretic operator from $\wp(W)^n \times W^m$ into $\wp(W)$.

We can obtain that operator by replacing in φ all connectives (Boolean and modal operators) by their respective set-theoretic counterparts. Usually, however, we will simply identify formulae with the operators they define.

DEFINITION 5.1
An $\mathcal{L}^n_{r(\tau)}$-formula $\varphi = \varphi(p_1, \dots, p_n, \mathbf{i}_1, \dots, \mathbf{i}_m)$ is a *closed operator on (reversive) descriptive frames*, if for every (reversive) descriptive frame $\mathfrak{F}$, $P_1, \dots, P_n \in \mathbf{C}(\mathbb{W})$, $w_1, \dots, w_n \in W$ only if $\varphi(P_1, \dots, P_n, \{w_1\}, \dots, \{w_n\}) \in \mathbf{C}(\mathbb{W})$. Further φ is a *closed formula on (reversive) descriptive frames*, if for every (reversive) descriptive frame $\mathfrak{F}$, $P_1, \dots, P_n, \in \mathbb{W}$ and $w_1, \dots, w_n \in W$ only if $\varphi(P_1, \dots, P_n, \{w_1\}, \dots, \{w_n\}) \in \mathbf{C}(\mathbb{W})$.

Similarly, an $\mathcal{L}^n_{r(\tau)}$-formula is an *open operator on (reversive) descriptive frames* if, whenever applied to open sets in a (reversive) descriptive frame, it produces an open set; it is an *open formula on (reversive) descriptive frames* if, whenever applied to admissible sets in such a frame, it produces an open set.

Note that for any $\alpha \in \mathrm{ML}_r(\tau)$, the operator $\langle\alpha\rangle$ distributes over arbitrary unions, and $[\alpha]$ distributes over arbitrary intersections. Since every closed set can be obtained as the intersection of admissible sets and every open set as the union of admissible sets, we have the following: on descriptive frames, $\langle\alpha\rangle(p_1, \dots, p_{\rho(\alpha)})$ is an open operator and $[\alpha](p_1, \dots, p_{\rho(\alpha)})$ is a closed operator for every $\alpha \in \mathrm{MT}_\tau$. On reversive descriptive frames we also have that every $\langle\alpha\rangle(p_1, \dots, p_{\rho(\alpha)})$ is an open operator and every $[\alpha](p_1, \dots, p_{\rho(\alpha)})$ is a closed operator, and that holds for any $\alpha \in \mathrm{MT}_{r(\tau)}$.

5.2 D-persistence in $\mathcal{L}_\tau$ and $\mathcal{L}_{r(\tau)}$

We extend ad hoc the notion of satisfiability of $\mathcal{L}^n_{r(\tau)}$-formulae in arbitrary (reversive) general τ-frames as follows: $\varphi \in \mathcal{L}^n_{r(\tau)}$ is satisfiable in a pointed general frame $(\mathfrak{F}, w) = ((W, \{R_\alpha\}_{\alpha\in\mathrm{MT}}, \mathbb{W}), w)$, if there is a valuation V assigning admissible sets (i.e. members of $\mathbb{W}$) to propositional variables and any singletons to nominals, such that $((\mathfrak{F}, V), w) \Vdash \varphi$. The notions of (parameterized) local and global satisfiability and validity are extended accordingly.

The monadic version of the next lemma goes back to [11]; the polyadic case is proven in [18].

LEMMA 5.2 (Esakia's Lemma for Diamonds)
Let $\mathfrak{F}$ be a descriptive τ-frame. Then for any downward-directed family $\{X_{1_i} \times \cdots \times X_{n_i} : i \in I\}$ of non-empty closed subsets of W^n, and any n-ary $\alpha \in \mathrm{MT}_\tau$, *it is the case that*

$$\bigcap_{i\in I} \langle\alpha\rangle(X_{1_i}, \dots, X_{n_i}) = \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \dots, \bigcap_{i\in I} X_{n_i}).$$

COROLLARY 5.3
For any n-ary $\alpha \in \mathrm{MT}_\tau$, $\langle\alpha\rangle(p_1, \dots, p_n)$ is a closed operator on descriptive τ-frames.

The results above apply likewise to every $\alpha \in \mathrm{MT}_{r(\tau)}$ in reversive descriptive frames. However, for the case of $\mathcal{L}_\tau$ we will need analogous results 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 α. 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 5.3 to $\langle\alpha\rangle$ for arbitrary $\alpha \in \mathrm{ML}_{\mathrm{r}(\tau)}$.

LEMMA 5.4
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 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.

We prove both claims by simultaneous induction on $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$. The base case is for basic modal terms $\alpha \in \tau$, and holds by Corollary 5.3 and the definition of descriptive τ-frames.

Now suppose that $\alpha, \beta_1, \ldots, \beta_n \in \mathrm{ML}_\tau$ 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 an 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_α^σ is tight in $\mathfrak{F}$. To that end, assume that $\neg R_\alpha^\sigma y_0, y_1, \ldots, y_n$. 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 exists $x_1 \in A_1, \ldots, x_n \in A_n$ 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 exists $x_1 \in A_{\sigma(1)}, \ldots, x_n \in A_{\sigma(n)}$ (i.e. $x_{\sigma(1)} \in A_1, \ldots, x_{\sigma(n)} \in A_n$) 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.[5]
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

$$\langle\alpha^\sigma\rangle(A_1, \ldots, A_n) = \bigcap\{B \in \mathbb{W} \mid \langle\alpha^\sigma\rangle(A_1, \ldots, A_n) \subseteq B\}. \tag{1}$$

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

$$R^\sigma_\alpha(y_0) \cap (A_1 \times \ldots \times A_n) = \emptyset. \tag{2}$$

This means that

$$\forall y_1 \ldots \forall y_n((y_1, \ldots, y_n) \in A_1 \times \cdots \times A_n \rightarrow (y_1, \ldots, y_n) \notin R^\sigma_\alpha(y_0)) \tag{3}$$

i.e.

$$\forall y_1 \ldots \forall y_n((y_1, \ldots, y_n) \in A_1 \times \cdots \times A_n \rightarrow \neg R^\sigma_\alpha y_0 y_1, \ldots, y_n, \tag{4}$$

i.e.

$$\forall y_1 \ldots \forall y_n((y_1, \ldots, y_n) \in A_1 \times \cdots \times A_n \rightarrow \neg R_\alpha y_{\sigma(0)}, y_{\sigma(1)} \ldots, y_{\sigma(n)}. \tag{5}$$

Note that in (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.

$$R_\alpha(y_{\sigma(0)}) = \bigcap\{-(-B_1 \times \cdots \times -B_n) \mid R_\alpha(y_{\sigma(0)}) \subseteq -(-B_1 \times \cdots \times -B_n), B_i \in \mathbb{W}\} \tag{6}$$

[5] The proof of this case generalizes, and closely follows, that of theorem 72 in [18], where the current theorem is proved for the more restricted case of modal terms in $MT_{\tau r}$ rather than $MT_{r(\tau)}$.

i.e.

$$R_\alpha(y_{\sigma(0)}) = \bigcap\{-(-B_1 \times \cdots \times -B_n) \mid y_{\sigma(0)} \in [\alpha](B_1, \ldots, B_n), B_i \in \mathbb{W}\}. \tag{7}$$

From (7) and (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

$$A_1 \times \cdots \times A_n \subseteq \bigcup\{(-B_{\overline{\sigma}(1)}^{\overline{y}}) \times \cdots \times [\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}}) \times \cdots \times (-B_{\overline{\sigma}(n)}^{\overline{y}}) \mid \overline{y} \in A_1 \times \cdots \times A_n\} \tag{8}$$

where $[\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}})$ is 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 $\bigcup\{(-B_{\overline{\sigma}(1)}^{\overline{y}}) \times \cdots \times [\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}}) \times \cdots \times (-B_{\overline{\sigma}(n)}^{\overline{y}}) \mid \overline{y} \in A_1 \times \cdots \times A_n\}$ 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

$$\begin{aligned} A_1 \times \cdots \times A_n \subseteq \bigcup\{(-B_{\overline{\sigma}(1)}^{\overline{y}}) \times \cdots \times [\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}}) \\ \times \cdots \times (-B_{\overline{\sigma}(n)}^{\overline{y}}) \mid \overline{y} \in A_1' \times \cdots \times A_n'\} \end{aligned} \tag{9}$$

By the monotonicity of $\langle\alpha^\sigma\rangle$ and its distributivity over arbitrary unions we have

$$\langle\alpha^\sigma\rangle(A_1, \ldots, A_n) \subseteq \bigcup\{\langle\alpha^\sigma\rangle(-B_{\overline{\sigma}(1)}^{\overline{y}}, \ldots, [\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}}), \ldots, -B_{\overline{\sigma}(n)}^{\overline{y}}) \mid \overline{y} \in A_1' \times \cdots \times A_n'\} \tag{10}$$

By the contrapositive of the implication proved in Lemma 2.2 we have, for each $\overline{y} \in A_1' \times \cdots \times A_n'$,

$$\langle\alpha^\sigma\rangle(-B_{\overline{\sigma}(1)}^{\overline{y}}, \ldots, [\alpha](B_1^{\overline{y}}, \ldots, B_n^{\overline{y}}), \ldots, -B_{\overline{\sigma}(n)}^{\overline{y}}) \subseteq B_j^{\overline{y}}. \tag{11}$$

Hence we have

$$\langle\alpha^\sigma\rangle(A_1, \ldots, A_n) \subseteq \bigcup\{B_j^{\overline{y}} \mid \overline{y} \in A_1' \times \cdots \times A_n'\} = B_0 \tag{12}$$

But then $B_0 \in \mathbb{W}$ and $y_0 \notin B_0$, and we are done. ⊣

By the duality of $\langle\alpha\rangle$ and $[\alpha]$, we obtain:

COROLLARY 5.5
For any n-ary modal term $\alpha \in \mathrm{MT}_{r(\tau)}$, it is the case that $[\alpha](p_1, \ldots, p_n)$ 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 5.6 (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

$$\langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \bigcap_{i\in I}\langle\alpha\rangle(X_{1_i}, \ldots, X_{n_i})$$

whenever $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i\in I}$ is a family of downward-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 $x_0 \notin \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \ldots, \bigcap_{i\in I} X_{n_i})$, i.e.

$$\{x_0\} \cap \langle\alpha\rangle(\bigcap_{i\in I} X_{1_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Hence

$$\bigcap_{i\in I} X_{1_i} \cap \langle\alpha\rangle^{-1}(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Now, since $\{x_0\}$ is closed in $T(\mathfrak{F})$, by Lemma 5.4 we have here a family of closed sets with empty intersection. By compactness, there is a finite subfamily with empty intersection, say

$$X_{1_{i_1}} \cap \cdots \cap X_{1_{im}} \cap \langle\alpha\rangle^{-1}(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Furthermore, since $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i\in I}$ is downward-directed, then, so is every family $\{X_{1_i}\}_{i\in I}, \ldots, \{X_{n_i}\}_{i\in I}$. Therefore, we can find $X_1 \in \{X_{1_i}\}_{i\in I}$ such that $X_1 \subseteq X_{1_{i_1}} \cap \cdots \cap X_{1_{i_m}}$, and hence

$$X_1 \cap \langle\alpha^{-1}\rangle(\{x_0\}, \bigcap_{i\in I} X_{2_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Equivalently, it must be the case that

$$\bigcap_{i\in I} X_{2_i} \cap \langle\alpha^{-2}\rangle(X_1, \{x_0\}, \bigcap_{i\in I} X_{3_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

In the same way as above, we find a $X_2 \in \{X_{2_i}\}_{i\in I}$ such that

$$X_2 \cap \langle\alpha^{-2}\rangle(X_1, \{x_0\}, \bigcap_{i\in I} X_{3_i}, \ldots, \bigcap_{i\in I} X_{n_i}) = \emptyset.$$

Proceeding likewise, we find $X_3 \in \{X_{3_i}\}_{i\in I}, \ldots, X_n \in \{X_{n_i}\}_{i\in I}$ such that

$$\{x_0\} \cap \langle\alpha\rangle(X_1, \ldots, X_j, .., X_n) = \emptyset.$$

Therefore,

$$x_0 \notin \bigcap_{i_1,\dots,i_n \in I} \langle\alpha\rangle(X_{1_{i_1}}, \dots, X_{n_{i_n}}).$$

The result follows once we note that, by the downward directedness of $\{X_{1_i} \times \cdots \times X_{n_i}\}_{i \in I}$,

$$\bigcap_{i_1,\dots,i_n \in I} \langle\alpha^{-j}\rangle(X_{1_{i_1}}, \dots, X_{n_{i_n}}) = \bigcap_{i \in I} \langle\alpha^{-j}\rangle(X_{1_i}, \dots, X_{n_i}). \qquad \dashv$$

Definition 5.7

(1) 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.
(2) By an *inverse diamond (box)* we mean a diamond (box) $\langle\alpha\rangle$ ($[\alpha]$) with $\alpha \in \mathrm{MT}_{\mathrm{r}(\tau)} - \mathrm{MT}_\tau$. A formula $\varphi \in \mathcal{L}^n_{r(\tau)}$ is *syntactically closed* if all occurrences of nominals and inverse diamonds in φ are positive, and all occurrences of inverse boxes in φ are negative; if the formula is in negation normal form, the latter simply means that it contains no inverse boxes. Likewise, φ is *syntactically open* if all occurrences of nominals and inverse diamonds in φ are negative, and all occurrences of inverse boxes in φ are positive. Clearly, maps syntactically open formulae to syntactically closed formulae, and vice versa.

Lemma 5.8

(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 $\langle\alpha\rangle$ and $[\alpha]$ are both open and closed operators for $\alpha \in \mathrm{MT}_\tau$, due to Corollary 5.3. In the second case, we also use the facts that by Lemma 5.4 and Corollary 5.5, $\langle\alpha\rangle$ and $[\alpha]$ are, respectively, closed and open operators on descriptive frames, for any $\alpha \in \mathrm{ML}_{\mathrm{r}(\tau)}$. $\dashv$

A notational convention: whenever we write a formula $\varphi(q_1, \dots, q_n, \mathbf{i}_1, \dots, \mathbf{i}_m)$ we assume that $\mathsf{PROP}(\varphi) \subseteq \{q_1, \dots, q_n\}$ and $\mathsf{NOM}(\varphi) \subseteq \{\mathbf{i}_1, \dots \mathbf{i}_m\}$.

Lemma 5.9
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 $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ be a descriptive τ-frame, such that one of the following holds:

(1) φ is syntactically closed;
(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 $\alpha \in \mathrm{ML}_{\mathrm{r}(\tau)} - \mathrm{ML}_\tau$. The inductive step for $\langle\alpha\rangle, \alpha \in \mathrm{MT}_{\mathrm{r}(\tau)}$ follows by Lemma 5.4. The inductive step for $[\alpha]$ is immediate from the fact that $[\alpha], \alpha \in \mathrm{ML}_\tau$, is a closed operator, as was noted earlier.

The proof for the second case is essentially the same, as the inductive step for $[\alpha], \alpha \in \mathrm{ML}_{\mathrm{r}(\tau)}$ is now the same as for $[\alpha], \alpha \in \mathrm{ML}_\tau$. ⊣

Lemma 5.10 (Esakia's Lemma for syntactically closed and 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 $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\mathrm{MT}}, \mathbb{W})$ be a descriptive τ-frame, such that one of the following holds:

(1) φ is syntactically closed;
(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 a downward-directed family of closed sets $\{C_i : i \in I\}$ 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, treated in [5].

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

$$\langle\alpha\rangle(\gamma_1(\bigcap_{i\in I} C_i), \dots, \gamma_n(\bigcap_{i\in I} C_i)) = \emptyset = \langle\alpha\rangle(\bigcap_{i\in I} \gamma_1(C_i), \dots, \bigcap_{i\in I} \gamma_n(C_i)),$$

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 5.9, $\{\gamma_1(C_i) \times \dots \times \gamma_n(C_i) : i \in I\}$ is a family of non-empty closed sets. Moreover, this family is downward-directed. For, consider any finite subset $\{\gamma_1(C_i) \times \dots \times \gamma_n(C_i)\}_{i=1,2,\dots,m}$ of $\{\gamma_1(C_i) \times \dots \times \gamma_n(C_i) : i \in I\}$. By the downward directedness of $\{C_i : i \in I\}$, there is a $C \in \{C_i : i \in I\}$ such that $C \subseteq \bigcap_{i=1}^m C_i$. But then, $\gamma_k(C) \in \{\gamma_k(C_i) : i \in I\}$ and $\gamma_k(C) \subseteq \bigcap_{i=1}^n \gamma_k(C_i)$ by the upwards monotonicity of γ 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 5.6 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 $\alpha \in \mathrm{ML}_{\mathrm{r}(\tau)}$. This case follows by the distributivity of $[\alpha]$ over arbitrary unions and the fact that the algebra $\mathbb{W}$ is now also closed under such $[\alpha]$. ⊣

Lemma 5.11 (Restricted version of Ackermann's Lemma for descriptive frames)
Let $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in\mathrm{MT}}, \mathbb{W})$ be a descriptive τ-frame and $A(q_1, \ldots, q_n, \mathbf{i}_1, \ldots, \mathbf{i}_m)$, $B(q_1, \ldots, q_n, p, \mathbf{i}_1, \ldots, \mathbf{i}_m) \in \mathcal{L}^n_{r(\tau)}$, be such that B is negative in p and one of the following holds:

(1) A is syntactically closed and B is syntactically open;
(2) A is nominal-positive, B is nominal-negative, and $\mathfrak{F}$ is reversive.

Then for all $Q_1, \ldots, Q_n \in \mathbb{W}$ and $x_1, \ldots, x_m \in W$:

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

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

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

Proof. For the sake of brevity we will suppress the parameters $Q_1, \ldots, Q_n, \{x_1\}, \ldots, \{x_m\}$ in what follows, and will simply write A, $B(P)$, etc.

We prove the first case—the proofs of both cases are completely analogous. The implication from bottom to top follows by the downward monotonicity of B in p. Now, suppose $B(A) = W$. Let $B'(p)$ be the negation of $B(p)$ written in negation normal form. Then $B'(p)$ is a syntactically closed formula, 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 5.8 that A is a closed subset of W and hence that $A = \bigcap\{C \in \mathbb{W} : A \subseteq C\}$. Hence $\emptyset = B'(A) = B'(\bigcap\{C \in \mathbb{W} : A \subseteq C\}) = \bigcap\{B'(C) : C \in \mathbb{W} \text{ and } A \subseteq C\}$, by Lemma 5.10. Again by Lemma 5.8, $\{B'(C) : C \in \mathbb{W}, A \subseteq C\}$ is a family of closed sets with empty intersection. Hence, by compactness, there must be a finite subfamily $C_1, \ldots, C_n$, 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 $P = C$. ⊣

Hereafter, we will refer to SQEMA-equations which are pure formulae (i.e. do not contain propositional variables) as *pure equations*, and to the rest, as *non-pure equations*. A straightforward inductive argument establishes the following facts:

Lemma 5.12

(1) During the entire (successful or unsuccessful) execution of SQEMA on any input formula from $\mathcal{L}_\tau$, all non-pure SQEMA-equations are syntactically open formulae.

(2) During the entire (successful or unsuccessful) execution of SQEMA on any input formula from $\mathcal{L}_{r(\tau)}$, all non-pure SQEMA-equations are nominal-negative formulae.

LEMMA 5.13
Let $\mathsf{Sys}_0, \ldots, \mathsf{Sys}_r$ be the sequence of systems of equations produced on one disjunctive branch by SQEMA when executed on a certain input formula $\varphi(q_1, \ldots, q_n)$ from $\mathcal{L}_\tau$ (respectively, $\mathcal{L}_{r(\tau)}$), and let $\mathbf{i}, \mathbf{i}_1, \ldots, \mathbf{i}_m$ be the nominals introduced during the execution. Then for any descriptive (respectively, reversive descriptive) τ-frame $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ and a current state $w \in W$, there are $Q_1, \ldots, Q_n \in \mathbb{W}$ and $x_1, \ldots, x_m \in W$ such that

$$\mathsf{Form}(\mathsf{Sys}_i)(Q_1, \ldots, Q_n, \{w\}, \{x_1\}, \ldots, \{x_m\}) = W$$

if and only of there are $Q_1, \ldots, Q_n \in \mathbb{W}$ and $x_1, \ldots, x_m \in W$ such that

$$\mathsf{Form}(\mathsf{Sys}_{i+1})(Q_1, \ldots, Q_n, \{w\}, \{x_1\}, \ldots, \{x_m\}) = W,$$

for $0 \leq i < r$.

PROOF. The proofs of both cases are analogous and will be done simultaneously. Note that $\mathsf{Form}(\mathsf{Sys}_0) = \neg\mathbf{i} \vee \neg\varphi$. For each i, the system Sys_{i+1} is obtained from the system Sys_i by the application of some transformation rule. We need to verify that whichever transformation rule was applied, $\mathsf{Form}(\mathsf{Sys}_i)$ is globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$ if and only if $\mathsf{Form}(\mathsf{Sys}_{i+1})$ is. This is immediate to see for all the transformation rules except the Ackermann-rule. So suppose that Sys_{i+1} is obtained from Sys_i by application of this rule. Then $\mathsf{Form}(\mathsf{Sys}_i) = \bigwedge_j (A_j \vee p) \wedge \bigwedge_j B_j \wedge \bigwedge_j C_j$, where no A_j contains p, each B_j is negative in p, and no C_j contains any occurrence of p. Note that all pure equations in the system Sys_i will be among the C_j. By Lemma 5.12, $\bigvee_j \neg A_j$ is syntactically closed (respectively, nominal-positive) and $\bigwedge_j B_j$ is syntactically open (respectively, nominal-negative).

Then $\mathsf{Form}(\mathsf{Sys}_{i+1}) = \bigwedge_j (B'_j) \wedge \bigwedge_j C_j$, where each B'_j is obtained from B_j by substituting $\bigwedge_j A_j$ for all occurrences of $\neg p$, which is semantically equivalent to substituting $\bigvee_j \neg A_j$ for all occurrences of p. The proof is complete once we appeal to Lemma 5.11. ⊣

THEOREM 5.14

(1) If SQEMA succeeds on an $\mathcal{L}_\tau$-formula φ, then φ is locally persistent with respect to the class of all descriptive τ-frames.
(2) If SQEMA succeeds on an $\mathcal{L}_{r(\tau)}$-formula φ, then φ is locally persistent with respect to the class of all reversive descriptive τ-frames.

PROOF. Again, the proofs of both cases are completely analogous.

First, we make the simplifying assumption that the execution does not branch into different systems because of the bubbled-up disjunctions in the negated input formula. This assumption is safe, since conjunctions of d-persistent formulae are d-persistent.

Let $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha \in \mathrm{MT}}, \mathbb{W})$ be a (reversive) descriptive τ-frame and $w \in W$. Then, $(\mathfrak{F}, w) \not\Vdash \varphi$ iff $\neg\mathbf{i} \vee \neg\varphi$ is globally $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}$. Note that $\|\neg\mathbf{i} \vee \neg\varphi$ is exactly the initial system of SQEMA-equations obtained when the algorithm is run on φ.

Recall that $\mathsf{pure}(\varphi)$ is the conjunction of the equations in the final pure system in the execution. So $\mathsf{pure}(\varphi) = \mathsf{Form}(\mathsf{Sys})$ for some system obtained from $\|\neg\mathbf{i} \vee \neg\varphi$ by the application of transformation rules. Hence, by Lemma 5.13, $\neg\mathbf{i} \vee \neg\varphi$ is globally $[\mathbf{i} := w]$-satisfiable in $\mathfrak{F}$ iff $\mathsf{pure}(\varphi)$ is so satisfiable.

Now, since we allow nominals to range over all singletons in general frames, $\mathsf{pure}(\varphi)$ is globally $[\mathbf{i} := w]$-satisfiable in $\mathfrak{F}$, iff it is $[\mathbf{i} := w]$-satisfiable in the underlying Kripke frame $\mathfrak{F}_\sharp$ of $\mathfrak{F}$.

By Lemma 4.2, this is the case iff $\neg\mathbf{i} \vee \neg\varphi$ is $[\mathbf{i} := w]$-satisfiable on $\mathfrak{F}_\sharp$ iff $(\mathfrak{F}_\sharp, w) \nVdash \varphi$.

Thus, we have proved that $(\mathfrak{F}, w) \Vdash \varphi$ iff $(\mathfrak{F}_\sharp, w) \Vdash \varphi$, whence the (reversive) d-persistence of φ. ⊣

6 Languages with nominals and di-persistence

To achieve completeness, hybrid modal logics usually need special additional rules of inference [15, 2] which, in a modified canonical model construction, guarantee that every world in the canonical model contains a nominal, and therefore every singleton is an admissible set in the canonical general frame. Thus, the so-constructed canonical general frame for a hybrid logic is *discrete*, and that is why we are now interested in hybrid modal formulae which are *di-persistent*, i.e. persistent with respect to discrete, rather than descriptive frames.

6.1 SQEMA^n *and di-persistence*

Again, we have to distinguish two cases: arbitrary and reversive hybrid languages. The case of $\mathcal{L}^n_{r(\tau)}$ is quite easy, as it does not require any modification of SQEMA because it preserves that input language, and hence the valuations referred to in Ackermann's lemma are admissible in every discrete reversive τ-frame. Thus, we have the following:

THEOREM 6.1
All formulae from $\mathcal{L}^n_{r(\tau)}$ on which SQEMA succeeds are locally persistent with respect to all discrete reversive τ-frames.

For the non-reversive case, here we will show how SQEMA can be modified by means of a simple restriction on the application of the Ackermann-rule, to guarantee that when it succeeds on an $\mathcal{L}^n_\tau$-formula, that formula is locally persistent with respect to the class of all (not necessarily reversive) discrete frames.

LEMMA 6.2 (Ackermann's Lemma for discrete τ-frames)
Let $\mathfrak{F} = (W, \{R_\alpha\}_{\alpha\in \mathrm{MT}_\tau}, \mathbb{W})$ be a discrete τ-frame and let $A \in \mathcal{L}^n_\tau$ and $B(p) \in \mathcal{L}^n_{r(\tau)}$ 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 $[\![A]\!]_\mathcal{M}$. ⊣

We now modify SQEMA to obtain SQEMA^n by restricting the scope of applications of the Ackermann-rule as follows:

Ackermann-rule on discrete frames: This rule is based on the equivalence given in Ackermann's Lemma for discrete frames.

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

where:

(1) p does not occur in $A_1, \ldots, A_n$;
(2) $A_1, \ldots, A_n \in \mathcal{L}^n_\tau$, i.e. these formulae contain no inverse modalities; and
(3) each of $B_1, \ldots, B_m$ is negative in p.

LEMMA 6.3
Let Sys be a system of SQEMA-equations, Sys$'$ be a system obtained from Sys by the application of a transformation rule of SQEMA, and $(\mathfrak{F}, w)$ be a pointed discrete frame. Then Form(Sys) is globally $[\mathbf{i} := w]$-satisfiable on $(\mathfrak{F}, w)$, if and only if *Form*(Sys$'$) is globally $[\mathbf{i} := w]$-satisfiable on $(\mathfrak{F}, w)$.

PROOF. It suffices to note that all transformation rules of SQEMA^n maintain this type of parameterized satisfiability on discrete frames, the case for the Ackermann-rule for discrete frames being justified by Lemma 6.2. ⊣

Let $\mathsf{Pure}_{\mathsf{SQEMA}^n}(\varphi)$ be the formula $\neg\mathsf{pure}(\varphi)$ where $\mathsf{pure}(\varphi)$ is the pure formula obtained in step Postprocessing.2 of the algorithm when successful on input φ. The proof of the next theorem is directly analogous to that of Theorem 4.3, appealing to Lemma 6.3, where the latter appeals to Lemma 4.2.

THEOREM 6.4 (Correctness of SQEMA^n on discrete frames)
If SQEMA succeeds on an input formula $\varphi \in \mathcal{L}^n_\tau$, then $\mathsf{Pure}_{\mathsf{SQEMA}^n}(\varphi)$ is locally equivalent to φ over the class of all discrete frames.

COROLLARY 6.5
Every input formula $\varphi \in \mathcal{L}^n_\tau$ on which SQEMA succeeds is (locally) di-persistent. Hence, if SQEMA succeeds on a given formula $\varphi \in \mathcal{L}^n_\tau$ then the logics $K^n_\tau \oplus \varphi$, $K^{n,@}_\tau \oplus \varphi$, and $K^{n,u}_\tau \oplus \varphi$ are strongly complete.

We will demonstrate the strength of SQEMA, by establishing some completeness results.

DEFINITION 6.6
A formula $\varphi \in \mathcal{L}^n_\tau$ 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 these occurrences are negative. Respectively, a formula $\varphi \in \mathcal{L}^n_\tau$ 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 $\varphi \in \mathcal{L}_\tau^n$ is diamond-uniform if, after transforming φ in 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 6.7

- Some diamond-uniform formulae:

$$\Diamond p \to \Box\Diamond p,\ \Diamond p \to \Diamond\Box p,\ \Diamond p \to \Diamond\Diamond p,\ [\mathbf{2}](p, p) \to p,\ \langle\mathbf{2}\rangle(p, q) \to [\mathbf{2}](\langle\mathbf{2}\rangle(p, \neg q), \langle\mathbf{2}\rangle(\neg q, p)).$$

- Some formulae which are not diamond-uniform:

$$\Box p \to \Diamond p,\ \Box p \to \Diamond\Box p,\ \Box p \to \Box\Diamond p,\ [\mathbf{2}](p, \neg p) \to p,\ \langle\mathbf{2}\rangle(p, q) \to [\mathbf{2}](\langle\mathbf{2}\rangle(p, q), \langle\mathbf{2}\rangle(q, \neg p)).$$

$\lhd$

Remark 6.8
Recall from [2] that 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 a Sahlqvist implication whose antecedent is a very simple Sahlqvist antecedent (while, the consequent is a positive formula). The very simple Sahlqvist formulae are probably the best known class of non-pure di-persistent formulae. Note that every very simple Sahlqvist formula is diamond-uniform, since every negative occurrence of a variable comes from the antecedent, and is hence not in the scope of any positive diamond.

Proposition 6.9
Every diamond-uniform formula in the basic modal language is locally equivalent (i.e. over the class of all pointed general frames) to a formula built up from very simple Sahlqvist formulae, by applying, $\wedge$, $\vee$, and boxes.

Proof. Let $\varphi \in \mathcal{L}_\tau$ be a diamond-uniform formula. We will prove the claim by showing how φ can be constructed, up to local equivalence, from very simple Sahlqvist formulae by using only conjunctions, disjunctions, and boxes. First, substitute $\bot$ for all variables in which φ is positive and $\top$ for all those in which it is negative. Then, rewrite in negation normal form, and let us call the resulting formula φ_0. Change the polarity of each propositional variable p (i.e. uniformly substitute $\neg p$ for p, and get the formula back in negation normal form by eliminating any double negations) which has a negative occurrence in φ_0 which is in the scope of a diamond, and call the resulting formula φ_1. Thus, in φ_1, no negative occurrence of a variable is in the scope of a diamond. Hence, φ_1 is built up from positive and negative formulas in negation normal forms in which no diamonds occur, using conjunctions, disjunctions, and boxes. The claim now follows when we note that: (i) rewriting a positive formula *Pos* as $\top \to Pos$ turns it into a very simple Sahlqvist formula, and (ii) if ψ is a negative formula in which no diamonds occur and $\gamma_1 \vee \cdots \vee \gamma_n$ is obtained by rewriting $\neg\psi$ in negation normal form and distributing diamonds and conjunctions over disjunctions as much as possible, then $(\gamma_1 \to \bot) \wedge \cdots \wedge (\gamma_1 \to \bot)$ is conjunctions of very simple Sahlqvist formulae, tautologically equivalent to ψ. $\dashv$

REMARK 6.10
Note that the class of formulae built up from very simple Sahlqvist formulae, by applying, $\wedge$, $\vee$, and boxes, introduced in Proposition 6.9, is exactly the class of formulae obtained by replacing 'Sahlqvist implications' with 'very simple Sahlqvist formulae' in the definition of Sahlqvist formulae given in [2], after relaxing the unnecessary [5] requirement that disjunctions are only applied to formulae not sharing variables.

THEOREM 6.11
SQEMA^n succeeds on all diamond-uniform formulae.

PROOF. We will refer to a system of SQEMA-equations as a *box-uniform system*, if it has the form

$$\left\|\begin{array}{l}\neg\mathbf{i}_1 \vee \psi_1\\ \vdots\\ \neg\mathbf{i}_n \vee \psi_n\end{array}\right. ,$$

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. Any propositional variable occurring in a box-uniform system of SQEMA-equations, 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 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 $\neg\mathbf{i}_i \vee p$. 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 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. $\dashv$

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.

COROLLARY 6.12
SQEMA^n succeeds on all very simple Sahlqvist formulae.

COROLLARY 6.13
All diamond-uniform formulae are di-persistent.

EXAMPLE 6.14
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. This 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)[6], while both fail on the tree itself, taken as a Kripke frame.

These examples show that the condition of diamond-uniformity, and respectively the restriction imposed on SQEMA^n are indeed essential. On the other hand, SQEMA^n fails on the formula $D = \Box p \to \Diamond p$ for no good reasons since it is di-persistent, and actually locally equivalent to the variable-free formula $\Diamond\top$. This failure can be prevented by adding suitable additional rules to SQEMA^n, to strengthen its propositional reasoning engine, e.g.:

$$\frac{A \vee C,\ \ A \vee \neg C}{A}$$

Enhanced with this rule, SQEMA^n will succeed on (the negation of) D, because after the application of the $\wedge$-rule and the $\Box$-rule we obtain

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

and then, by applying the new rule above, we can eliminate p and obtain $\| \Box^{-1}\neg\mathbf{i}$, which, after negation and simplification, produces the seriality formula: $\exists y Rxy$. ◁

6.2 *Adding the universal modality and the satisfaction operator*

Hybrid languages usually use either the universal modality or the satisfaction operator to empower the nominals. Of course, the universal modality can be treated like any other modality, and the algorithm will remain correct, but it could be naturally strengthened if special additional transformation rules are added to capture the axioms of the universal modality, i.e. S5 plus the inclusion axiom $[\mathsf{U}]p \to \Box p$ (in the monadic case). We, however, will not present this extension here, but will defer it to a sequel work where SQEMA will be customized to work on special classes of frames, e.g. on all transitive frames. As for the satisfaction operator, it is well-known that it can be expressed in two different ways by means of the universal modality: $@_c p \equiv \langle \mathsf{U} \rangle (c \wedge p) \equiv [\mathsf{U}](c \to p)$. Using these equivalences, the transformation rules for $[\mathsf{U}]$ can be converted into transformation rules for $@$.

7 Concluding remarks and further work

In this study, we have extended the core algorithm SQEMA introduced in [5] to arbitrary and reversive hybrid polyadic modal languages, in order to compute first-order equivalents and to establish d-persistence and di-persistence of the input modal formulae.

[6] This general frame was used in [25] to show the incompleteness of a certain hybrid logic involving the Church-Rosser formula $\Diamond\Box p \to \Box\Diamond p$.

We have considered two different reversive extensions of a polyadic modal language: the one-step extension $\mathcal{L}_{\tau r}$ and the completely reversive extension $\mathcal{L}_{r(\tau)}$. This could be avoided if, instead, we add to the basic language *transposers*: operators θ_{ij} that swap the i-th and j-th argument of a modal term, i.e. $\langle\theta_{ij}(\alpha)\rangle(A_1, \ldots, A_i, \ldots, A_j, \ldots, A_n) = \langle\alpha\rangle(A_1, \ldots, A_j, \ldots, A_i, \ldots, A_n)$. In the presence of transposers the languages $\mathcal{L}_{\tau r}$ and $\mathcal{L}_{r(\tau)}$ become equivalent, because, e.g. $(\alpha^{-j})^{-k}$ becomes equivalent to $(\theta_{jk}(\alpha))^{-k}$ and to $\theta_{jk}(\alpha^{-j})$. Clearly, the application of transposers preserves all important properties (tightness, closedness, etc.) of the modal terms, and so they can be used to simplify some of the proofs in Section 5.

Some further questions arising from this work include:

- Can the restriction imposed in SQEMA^n be lifted or weakened while still preserving di-persistence in non-reversive hybrid languages?
- Alternatively, can the completeness of SQEMA^n with respect to diamond-uniform formulae be extended to a larger class?

As it is evident from the results in this study, the algorithm SQEMA is quite powerful, but it is still amenable to various further strengthenings, which will be treated in sequel works. Besides the one mentioned in Section 6.2, here are the main directions for extension of SQEMA:

- The Ackermann-rule can be strengthened to test for monotonicity, rather than polarity, of the equations in the variable which is to be eliminated. The variants of employing semantic versions of SQEMA the Ackermann-rule are introduced and studied in [6].
- The scope of application of can also be extended by generalizing the Ackermann-rule as in [27, 28] to deal with the complex formulae introduced in [26]. Another extension of the Ackermann-rule is its recursive version which allows computing equivalents of modal formulae in FO + least fixed points. These extensions are introduced and studied in [7]. See also [23] for a recursive version of Ackermann's lemma. An extended modal recursive version of the Ackermann's lemma, suitable for complex-like formulae from [26], was given in [29].

Acknowledgements

We thank the anonymous referee for careful reading and some corrections. V.G. would like to acknowledge the financial support from the National Research Foundation of South Africa, through research grant GUN 2072436. D.V. was supported by the Bulgarian Ministry of Science and Education, Project NIP-1510.

References

[1] W. Ackermann. Untersuchung uber das Eliminationsproblem der mathematischen Logic. *Mathematische Annalen*, **110**, 390–413, 1935.

[2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[3] L. A. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, **56**, 1261–1272, 1991.

[4] W. Conradie, V. Goranko, and D. Vakarelov. Elementary canonical formulae: a survey on syntactic, algorithmic, and model-theoretic aspects. In *Advances in Modal Logic*,

R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, eds, vol. 5, pp. 17–51. Kings College London, 2005.

[5] 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**, 2006.

[6] W. Conradie and V. Goranko. Algorithmic correspondence and completeness in modal logic III: semantic extension of the algorithm SQEMA (in preparation).

[7] W. Conradie, V. Goranko, and D. Vakarelov. Algorithmic correspondence and completeness in modal logic IV: recursive and complex extensions of the algorithm SQEMA (in preparation).

[8] 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.

[9] P. Doherty, W. Lukaszewicz, and A. Szalas. Computing circumscription revisited. *Journal of Automated Reasoning*, **18**, 297–336, 1997.

[10] T. Engel. *Quantifier Elimination in Second-order Predicate Logic*. Diploma Thesis. MPI, Saarbrucken, 1996.

[11] L. L. Esakia. Topological Kripke Models. *Soviet Mathematics Doklady*, **15**, 147–151, 1974.

[12] D. Gabbay and H.-J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, **7**, 35–43, 1992.

[13] D. Georgiev, T. Tinchev, and D. Vakarelov. SQEMA – an algoritm for computing first-order equivalents in modal logic: a computer realization. In *Pioneers of Bulgarian Mathematics*, International Conference dedicated to Nicola Obrechkoff and Lubomir Tschakaloff, July 8–10. Sofia, 2006 (abstracts).

[14] J. Gustafsson. An implementation and optimization of an algorithm for reducing formulae in second-order logic. *Technical Report LiTH-MAT-R-96-04*. Department of Mathematics, Linkoping University, Sweden, 1996.

[15] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, **22**, 607–636, 1993.

[16] V. Goranko and D. Vakarelov. Sahlqvist Formulas Unleashed in Polyadic Modal Languages. In *Advances in Modal Logic,* F. Wolter *et al.*, eds, vol. 3, pp. 221–240. World Scientific, Singapore, 2002.

[17] V. Goranko and D. Vakarelov. Sahlqvist Formulae in Hybrid Polyadic Modal Languages. *Journal of Logic and Computation*, **11**, 737–754, 2001.

[18] V. Goranko and D. Vakarelov. Elementary Canonical Formulae: extending Sahlqvist Theorem. *Annals of Pure and Applied Logic*, **141**, 180–217, 2006.

[19] V. Goranko, U. Hustadt, R. Schmidt, and D. Vakarelov. SCAN is complete for all Sahlqvist formulae. In *Relational and Kleene-Algebraic Methods in Computer Science (Proc. of RelMiCS 7), LNCS 3051*, pp. 149–162. Springer, 2004.

[20] A. Nonnengart and A. Szzalas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In *Logic at Work, Essays dedicated to the memory of Helena Rasiowa*, E. Orlowska ed., pp. 89–108. Springer Physica-Verlag.

[21] A. Nonnengart, H.-J. Ohlbach, and A. Szalas. Elimination of predicate quantifiers. In *Logic, Language and Reasoning: Essays in honour of Dov Gabbay*, H. J. Ohlbach and U. Reyle, eds, pp. 159–181. Kluwer, 1999, Part I.

[22] A. Szalas. On the correspondence between modal and classical logic: an automated approach. *Journal of Logic and Computation*, **3**, 605–620, 1993.
[23] A. Szalas. Second-order quantifier elimination in modal contexts, In S. Flesca and G. Ianni, eds. JELIA'02, LNAI 2424, pp. 223–232. Springer-Verlag, 2002.
[24] H. Sahlqvist. Correspondence and completeness in the first and second-order semantics for modal logic. In *Proc. of the 3rd Scandinavial Logic Symposium, Uppsala 1973*, S. Kanger ed., pp. 110–143. North-Holland, Amsterdam, 1975.
[25] B. ten Cate, M. Marx, and P. Viana. Hybrid logics with Sahlqvist axioms. *Logic Journal of the IGPL*, **13**, 293–300, 2005.
[26] D. Vakarelov. Modal definability in languages with a finite number of propositional variables, and a new extention of the Sahlqvist class. In *Advances in Modal Logic*, P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev, eds, vol. 4, pp. 495–518. King's College Publications, London, 2003.
[27] D. Vakarelov. On a generalization of the Ackermann's Lemma for computing first-order equivalents of modal formulae. In *TARSKI Workshop*. Toulouse, France, March 11–13, 2003.
[28] D. Vakarelov. Modal definability, solving equations in modal algebras and generalization of the Ackermann Lemma. In *the Proceedings of 5th Panhellenic Logic Symposium*. Athens, pp. 182–189, July 25–28, 2006.
[29] D. Vakarelov. Solving recursive equations in complete modal algebras with applications to modal definability. In *Pioneers of Bulgarian Mathematics*. International Conference dedicated to Nicola Obrechkoff and Lubomir Tschakaloff, July 8–10. Sofia, 2006 (abstracts).
[30] J. F. A. K. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Napoli, 1983.

Received 1 February 2006