Algorithmic correspondence and completeness in modal logic.
IV. Semantic extensions of SQEMA

Willem Conradie[*] — Valentin Goranko[**]

[*] *Department of Mathematics, University of Johannesburg*
PO Box 524, Auckland Park, 2006, South Africa
wconradie@uj.ac.za

[**] *School of Mathematics, University of the Witwatersrand*
Private Bag 3, WITS 2050, Johannesburg, South Africa
goranko@maths.wits.ac.za

ABSTRACT. *In [CON 06b] we introduced the algorithm SQEMA for computing first-order equivalents and proving canonicity of modal formulae, and thus established a very general correspondence and canonical completeness result. SQEMA is based on transformation rules, the most important of which employs a modal version of a result by Ackermann that enables elimination of an existentially quantified predicate variable in a formula, provided a certain negative polarity condition on that variable is satisfied. In this paper we develop several extensions of SQEMA where that syntactic condition is replaced by a semantic one, viz. downward monotonicity. For the first, and most general, extension SemSQEMA we prove correctness for a large class of modal formulae containing an extension of the Sahlqvist formulae, defined by replacing polarity with monotonicity. By employing a special modal version of Lyndon's monotonicity theorem and imposing additional requirements on the Ackermann rule we obtain restricted versions of SemSQEMA which guarantee canonicity, too.*

KEYWORDS: *Modal correspondence and completeness, algorithm SQEMA, Sahlqvist formulae, inductive formulae, Lyndon monotonicity, canonicity.*

Dedicated to our teacher and collaborator Dimiter Vakarelov, on the occasion of 70th anniversary of his birthday.

Introduction

Every modal formula defines a second-order condition on Kripke frames. Yet, as is well known, may modal formulae actually characterize first-order definable frame classes. We will call such formulae *elementary*. Because of the great computational and theoretical advantages of first-order over second-order logic it is desirable to identify as large as possible classes of elementary modal formulae. Unfortunately, by Chagrova's Theorem ([CHA 91]), the elementarity problem for modal formulae is algorithmically undecidable. Thus, if the classes of elementary formulae we are interested in are to be decidable, we have to content ourselves with approximations.

Several such approximating classes are known from the literature. Membership is often specified in terms of syntactic shape as, for example, in the case of the Sahlqvist ([SAH 75]) and inductive formulae ([GOR 06b]). Another approach to the determination of a classes of elementary formulae is algorithmic. For example, one can feed the second-order translation of a modal-formula to a second-order quantifier elimination algorithm like SCAN ([GAB 92, GOR 03]) or DLS ([SZA 93, CON 06a]).

The most recent method for computing first-order frame correspondents for modal formulae is based on the algorithm SQEMA, introduced in [CON 06b]. The name SQEMA is an acronym for *Second-order Quantifier Elimination in Modal logic using Ackermann's Lemma*. This algorithm works directly with the modal syntax, thus eliminating any translation into second order logic. It was shown in [CON 06b] that SQEMA successfully computes first-order frame equivalents for all Sahlqvist and inductive formulae. Perhaps the most interesting feature of SQEMA is the fact that it is also an *automated completeness prover*, since every formula on which it succeeds is provably canonical. In [CON 06c] SQEMA was extended to polyadic and hybrid languages.

The core of the SQEMA engine is a transformation rule based on a modal version of Ackermann's Lemma [ACK 35] stated further. In the current paper we consider three extensions of SQEMA based on a more general version of this lemma, where the syntactic notion of negativity is replaced with its semantic correlate — monotonicity. This, coupled with the fact that monotonicity is an effectively decidable property of modal formulae, immediately yields a semantic version SemSQEMA of SQEMA with a significantly enlarged scope of applicability. In particular, we introduce a new 'semantic' generalization of the Sahlqvist formulae and show that SemSQEMA successfully computes first-order equivalents for these formulae. However, we cannot claim that the most general semantic version SemSQEMA of SQEMA guarantees canonicity, and in the rest of the paper we develop more involved variants of SemSQEMA for which canonicity can be proved.

The paper is organized as follows. After providing some preliminaries, in section 2 we introduce the algorithm SemSQEMA based on the semantic version of Ackermann's Lemma and illustrate it with some examples. In section 3 we present the general framework of the canonicity proof for the SQEMA-reducible formulae and show why this proof fails for SemSQEMA. In section 4 we study the relationship between the monotonicity and polarity of formulae in more detail, and prove versions of Lyndon's monotonicity theorem which respect the specific syntactic shapes of formulae needed to make the canonicity proof work. In section 5 we introduce and study a modification SemClsSQEMA of SemSQEMA in which the application of the rule based on Ackermann's Lemma is slightly restricted. This restriction enables us to prove that all SemClsSQEMA-formulae are canonical. We then show that all semantic Sahlqvist formulae are SemClsSQEMA-reducible and hence canonical. In section 6 we introduce another variant, SemRepSQEMA, which requires the explicit replacement of monotone formulae with positive ones and eventually guarantees canonicity, too. In the last section we outline an algorithm for more efficient computation of such positive equivalents working under some natural additional assumptions. In a concluding section we discuss briefly some open questions related to the semantic approach presented here.

1. Preliminaries

In this section we collect some basic definition and notations. Any undefined terms are as in [BLA 01]. We assume countably infinite disjoint sets of *propositional variables* and *nominals* PROP and NOM, respectively. The member of $\mathsf{AT} := \mathsf{PROP} \cup \mathsf{NOM}$ will be referred to as *atoms*. The language $\mathcal{L}_r^n$ is given by the abstract syntax $\varphi ::= \bot \mid p \mid \mathbf{i} \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi \mid \Diamond^{-1}\varphi$ for $p \in \mathsf{PROP}$ and $\mathbf{i} \in \mathsf{NOM}$. The sublanguages $\mathcal{L}_r$, $\mathcal{L}^n$ and $\mathcal{L}$ are obtained by omitting the clauses for $\mathbf{i}$, $\Diamond^{-1}$, both, respectively. The boolean connectives $\rightarrow$, $\wedge$ and $\leftrightarrow$ are defined as usual, and as usual $\Box\varphi := \neg\Diamond\neg\varphi$ and $\Box^{-1}\varphi := \neg\Diamond^{-1}\neg\varphi$. We write $\mathsf{PROP}(\varphi)$, $\mathsf{NOM}(\varphi)$, and $\mathsf{AT}(\varphi)$ for the sets of propositional variables, nominals, and atoms, respectively, occurring in φ. By writing $\varphi(\overline{a})$ we mean that $\mathsf{AT}(\varphi) \subseteq \overline{a}$, where $\overline{a}$ is a vector (or vectors) of atoms. Formula φ is *pure* if $\mathsf{AT}(\varphi) \subseteq \mathsf{NOM}$.

A formula is in *negation normal form* if it is written without the use of the connectives $\rightarrow$ and $\leftrightarrow$, and the negation sign appears only directly in front of atoms. An occurrence of an atom a in a formula φ is *positive* (*negative*) if it is in the scope of an even (odd) number of negations. φ is *positive* (*negative*) in a if all occurrences of a in φ are positive (negative).

A *Kripke frame* is a pair $\mathfrak{F} = (W, R)$ with W a non-empty set and $R \subseteq W^2$ a binary relation on W. A *Kripke model* based on a frame $\mathfrak{F} = (W, R)$ is a pair $\mathcal{M} = (\mathfrak{F}, V)$ with V a *valuation* assigning to every $p \in \mathsf{PROP}$ a set $V(p) \subseteq W$ where it is true, and to every $\mathbf{i} \in \mathsf{NOM}$ a singleton subset $V(\mathbf{i})$ of W where it is true. Models $\mathcal{M} = (W, R, V_M)$ and $\mathcal{N} = (W, R, V_N)$ based on the same frame $\mathfrak{F} = (W, R)$ are called p-variants, denoted $\mathcal{M} \sim_p \mathcal{N}$, if $V_M(q) = V_N(q)$ for all $q \in \mathsf{AT} - \{p\}$.

The *truth* of an $\mathcal{L}_r^n$-formula φ at a point m in a Kripke model $\mathcal{M}$, denoted $(\mathcal{M}, m) \Vdash \varphi$, is defined as usual. Particularly, $(\mathcal{M}, m) \Vdash \Diamond\varphi$ iff there is a point $n \in W$ such that Rmn and $(\mathcal{M}, n) \Vdash \varphi$, and $(\mathcal{M}, m) \Vdash \Diamond^{-1}\varphi$ iff there is a point $n \in W$ such that Rnm and $(\mathcal{M}, n) \Vdash \varphi$.

Based on this truth definition a valuation V can be extended from atoms to all formulae in a unique way. We will accordingly write $V(\varphi)$ for $\{m \in W \mid (\mathcal{M}, m) \Vdash \varphi\}$ when $\mathcal{M} = (W, R, V)$ is understood. We write $\mathcal{M} \Vdash \varphi$ if φ is true at every point in $\mathcal{M}$. Similarly we write $(\mathfrak{F}, m) \Vdash \varphi$ and say φ is *valid* at m in $\mathfrak{F}$ if $(\mathcal{M}, m) \Vdash \varphi$ for every model $\mathcal{M}$ based on $\mathfrak{F}$, and write $\mathfrak{F} \Vdash \varphi$, saying φ is *valid* on $\mathfrak{F}$, if $\mathcal{M} \Vdash \varphi$ for all models $\mathcal{M}$ based on $\mathfrak{F}$. These notations are extended to sets of formulas in the usual way.

Given a set Γ of formulas and a formula φ, we write $\Gamma \Vdash \varphi$ if φ is true at every point in every model where all members of Γ are true. We write $\Gamma \Vdash^{mod} \varphi$ if $\mathcal{M} \Vdash \Gamma$ implies $\mathcal{M} \Vdash \varphi$ for all models $\mathcal{M}$.

Two modal formulae φ and ψ are *semantically equivalent*, denoted $\varphi \equiv_{sem} \psi$, if they are true at the same points in any Kripke model. Further, φ and ψ are *model equivalent*, denoted $\varphi \equiv_{mod} \psi$, if $\mathcal{M} \Vdash \varphi$ iff $\mathcal{M} \Vdash \psi$, for all models $\mathcal{M}$.

Define L_0 to be the first-order language with $=$, a binary relation symbol R, and disjoint sets of individual variables $\mathsf{VAR} = \{x_0, x_1, \ldots\}$ and $\{y_{\mathbf{i}} \mid \mathbf{i} \in \mathsf{NOM}\}$. Also, let L_1 be the extension of L_0 with a sets of unary predicates $\{P_0, P_1, \ldots\}$ corresponding to the propositional variables in PROP. $\mathcal{L}$-formulae are translated into L_1 by means of the usual *standard translation* function $\mathrm{ST}(\cdot,\cdot)$. Recall that $\mathrm{ST}(\varphi, x)$ is defined by induction on φ. Particularly $\mathrm{ST}(\mathbf{i}, x) := y_{\mathbf{i}} = x$ for every $\mathbf{i} \in \mathsf{NOM}$ and $\mathrm{ST}(\Diamond\varphi, x) := \exists y(Rxy \wedge \mathrm{ST}(\varphi, y))$, where y is the first variable in VAR not appearing in $\mathrm{ST}(\varphi, x)$.

Of course, a Kripke model is nothing but an L_1-structure and a Kripke frame nothing but an L_0-structure. Indeed, we have for any model $\mathcal{M}$ and any formula $\varphi \in \mathcal{L}_r^n$, that $(\mathcal{M}, m) \Vdash \varphi$ iff $\mathcal{M} \models \mathrm{ST}(\varphi, x)[x := m]$. Similarly, any frame $\mathfrak{F}$, $(\mathfrak{F}, m) \Vdash \varphi$ iff $\mathfrak{F} \models \forall\overline{P}\forall\overline{y}\mathrm{ST}(\varphi, x)[x := m]$ where $\overline{P}$ is the vector of all predicates corresponding to propositional variables and $\overline{y}$ that of all variables corresponding to nominals occurring in φ.

A first-order formula $\alpha(x) \in L_0$ with one free variable is a *local frame correspondent* for a formula $\varphi \in \mathcal{L}_r^n$ if, for any Kripke frame $\mathfrak{F}$ and point w in $\mathfrak{F}$, it holds that $(\mathfrak{F}, w) \Vdash \varphi$ iff $\mathfrak{F} \models \alpha[x := w]$.

A *general frame* $\mathfrak{g} = (W, R, \mathbb{W})$ is the augmentation of a Kripke frame $\mathfrak{F} = (W, R)$ with an algebra $\mathbb{W}$ of subsets of W (called *admissible subsets*) which is closed under the boolean operations and under the operation $\langle R\rangle(X) = \{y \in W \mid Ryx \text{ for some } x \in X\}$. Note that we do *not* require closure under $\langle R^{-1}\rangle$. A *model based on a general frame* $\mathfrak{g} = (W, R, \mathbb{W})$ is a model (W, R, V) with V an *admissible valuation*, i.e., $V(a) \in \mathbb{W}$ for all $a \in \mathsf{AT}$. $\mathfrak{g}_\sharp = (W, R)$ is the *underlying Kripke frame* of $\mathfrak{g} = (W, R, \mathbb{W})$. A formula is *persistent* with respect to a class $\mathfrak{C}$ of general frames if for all $\mathfrak{g} \in \mathfrak{C}$, $\mathfrak{g} \Vdash \varphi$ implies $\mathfrak{g}_\sharp \Vdash \varphi$.

We will often identify $\mathcal{L}_r^n$-formulae and the *operators* defined by them on the (powersets of) the domains of (general) frames. That is to say, for $\varphi(\overline{a}) \in \mathcal{L}_r^n$, $\mathfrak{g} = (W, R, \mathbb{W})$ a general frame, and $\overline{X}$ a tuple of subsets of W we write $\varphi(\overline{X})$ for $V(\varphi)$ in $(\mathfrak{g}, V)$ where V is any (possibly non-admissible) valuation assigning $\overline{X}$ to $\overline{a}$.

With every general frame $\mathfrak{g} = (W, R, \mathbb{W})$ we associate the topological space $(W, T(\mathfrak{g}))$ where $T(\mathfrak{g})$ is the topology having $\mathbb{W}$ as a basis of clopen sets. The set of all closed sets (with respect to $T(\mathfrak{g})$) is denoted by $\mathsf{Cls}(\mathfrak{g})$. We further write $\mathsf{Sgl}(\mathfrak{g})$ for the set $\{\{w\} \mid w \in W\}$ of all singleton subsets of W.

DEFINITION 1. — *A general frame* $\mathfrak{g} = (W, R, \mathbb{W})$ *is said to be:*

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

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

compact *if every family of admissible sets from* $\mathbb{W}$ *with the finite intersection property (FIP) has non empty intersection (equivalently, if* $T(\mathfrak{g})$ *is compact). Recall that a family of sets has FIP if any finite subfamily has non-empty intersection;*

descriptive *if it is differentiated, tight and compact.*

The usual way of proving canonicity of a modal formula is to show that is persistent with respect to the class of descriptive general frames, or *d-persistent*, for short.

2. Ackermann's Lemma and a 'semantic' extension of SQEMA

The core result on which the original SQEMA-algorithm [CON 06b], as well as the extensions we introduce in this paper, are based, is a modal version of a lemma by Ackermann (Lemma 3). In this section we show how the algorithmic potential of this lemma can be better exploited than it has been in previous papers on SQEMA, by noting that, in the formulation of the lemma, the syntactic notion of *negativity* can be replaced by the more general *semantic* notion of *monotonicity*, and that this latter semantic property is amenable to algorithmic treatment. This leads to a generalized version of SQEMA which we will introduce and call SemSQEMA. We will illustrate SemSQEMA with examples and by developing a new extension of the class of Sahlqvist formulae, the members of which are SemSQEMA-reducible.

2.1. *Ackermann's Lemma*

DEFINITION 2. — *A formula* $\varphi \in \mathcal{L}_r^n$ *is said to be* upward monotone *(respectively,* downward monotone*) in a propositional variable* p*, if* $V(\varphi) \subseteq V'(\varphi)$ *whenever* $\mathcal{M} = (W, R, V) \sim_p \mathcal{M}' = (M, R, V')$ *and* $V(p) \subseteq V'(p)$ *(respectively,* $V'(p) \subseteq V(p)$*).*

$\varphi \in \mathcal{L}_r^n$ *is said to be* globally upward monotone *(respectively,* globally downward monotone*) in a propositional variable* p*, if* $\mathcal{M} \Vdash \varphi$ *implies* $\mathcal{M}' \Vdash \varphi$ *whenever* $\mathcal{M} = (W, R, V) \sim_p \mathcal{M}' = (W, R, V')$ *and* $V(p) \subseteq V'(p)$ *(respectively,* $V'(p) \subseteq V(p)$*).*

It is easy to see (by induction on φ) that the positivity (negativity) of φ in p is sufficient, but not necessary, for its upward (downward) monotonicity in p. Further monotonicity clearly implies global monotonicity, but not conversely. Indeed, $(p \wedge \Box\neg p)$ is globally upward monotone in p but not upward monotone in p.

The following, taken from [CON 06b], is a modal analogue of the lemma first proved by Ackermann in [ACK 35]. For another version see [SZA 93]. For convenience of presentation this lemma is often formulated in a more restricted way, by replacing downward monotonicity with negativity.

LEMMA 3 (MODAL ACKERMANN LEMMA). — *Let $A, B(p)$ be $\mathcal{L}_r^n$-formulae such that the propositional variable p does not occur in A and $B(p)$ is globally downward monotone in p. Then for any model $\mathcal{M}$, it is the case that $\mathcal{M} \Vdash B(A)$ iff $\mathcal{M}' \Vdash (A \rightarrow p) \wedge B(p)$ for some $\mathcal{M}' \sim_p \mathcal{M}$.*

PROOF. — Let $\mathcal{M} = (W, R, V)$. If $\mathcal{M} \Vdash B(A)$, then $\mathcal{M}' \Vdash (A \rightarrow p) \wedge B(p)$ for the model $(W, R, V') = \mathcal{M}' \sim_p \mathcal{M}$ such that $V'(p) = V(A)$. Conversely, if $\mathcal{M}' \Vdash (A \rightarrow p) \wedge B(p)$ for some model $\mathcal{M}' \sim_p \mathcal{M}$, then $\mathcal{M}' \Vdash B(A/p)$ since $B(p)$ is downwards monotone in p. Therefore, $\mathcal{M} \Vdash B(A/p)$. ■

The proof of the next lemma is straightforward.

LEMMA 4. — *An $\mathcal{L}_r^n$-formula $\varphi(p)$ is downwards monotone in p iff*

$$\Vdash \varphi(p) \rightarrow \varphi(p \wedge q)$$

where q is any variable not occurring in $\varphi(p)$.

Hence, the question of the monotonicity of an $\mathcal{L}_n^r$-formula in a propositional variable can be effectively reduced to the question of the validity of a related $\mathcal{L}_n^r$-formula, a problem which is decidable and EXPTIME-complete (see [ARE 00]). (By the way, note that testing validity is effectively reducible to testing monotonicity: $\Vdash \varphi$ iff $q \rightarrow \varphi$ is upwards monotone in q, where q is a variable not occurring in φ.)

It follows, that the applicability of Lemma 3 can be effectively determined in EXPTIME. In this paper we explore some consequences of that simple insight. In particular we develop 'semantic' versions of SQEMA. The word *semantic* indicates the fact that we have exchanged the syntactic property of negative/positive polarity for its semantic correlate — monotonicity.

2.2. *The algorithm SemSQEMA*

Some terminology — an expression of the form $\varphi \Rightarrow \psi$ with $\varphi, \psi \in \mathcal{L}_r^n$ is called a *SQEMA-sequent*[1], with φ and ψ the *antecedent* and *consequent* of the sequent, respectively. A finite set of SQEMA-sequents is called a *SQEMA-system*. We set $\mathsf{Form}(\varphi \Rightarrow \psi) := \neg\varphi \vee \psi$ and, for a system Sys, we let $\mathsf{Form}(\mathsf{Sys})$ be the conjunction of all $\mathsf{Form}(\varphi_i \Rightarrow \psi_i)$ for all sequents $\varphi_i \Rightarrow \psi_i \in \mathsf{Sys}$.

1. In [CON 06b] sequents are called 'equations' because of the analogy with solving systems of linear equations.

Table 1. *SemSQEMA Transformation Rules*

Rules for connectives

$$\frac{C \Rightarrow (A \wedge B)}{C \Rightarrow A, C \Rightarrow B} \quad (\wedge\text{-rule}) \qquad \frac{\mathbf{j} \Rightarrow \Diamond A}{\mathbf{j} \Rightarrow \Diamond \mathbf{k},\ \mathbf{k} \Rightarrow A} \quad (\Diamond\text{-rule}^*)$$

$$\frac{C \Rightarrow (A \vee B)}{(C \wedge \neg A) \Rightarrow B} \quad (\text{left-shift } \vee\text{-rule}) \qquad \frac{(C \wedge A) \Rightarrow B}{C \Rightarrow (\neg A \vee B)} \quad (\text{right-shift } \vee\text{-rule})$$

$$\frac{A \Rightarrow \Box B}{\Diamond^{-1} A \Rightarrow B} \quad (\Box\text{-rule}) \qquad \frac{\Diamond^{-1} A \Rightarrow B}{A \Rightarrow \Box B} \quad (\text{inverse } \Diamond\text{-rule})$$

$*$where $\mathbf{k}$ is a new nominal not occurring in the system.

Polarity switching rule

Substitute $\neg p$ for every occurrence of p in the system.

(Semantic) Ackermann-rule

The system

$$\left\| \begin{array}{l} A_1 \Rightarrow p \\ \vdots \\ A_n \Rightarrow p \\ B_1(p) \\ \vdots \\ B_m(p) \\ C \end{array} \right.$$

is replaced by

$$\left\| \begin{array}{l} B_1((A_1 \vee \ldots \vee A_n)/p) \\ \vdots \\ B_m((A_1 \vee \ldots \vee A_n)/p) \\ C \end{array} \right.$$

where:

1) p does not occur in $A_1, \ldots, A_n$ or C;

2) $\mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$ is downwards monotone in p.

Given a formula $\varphi \in \mathcal{L}$ as input, SemSQEMA processes it in three phases, with the goal to reduce φ first to a suitably equivalent pure, and then first-order formula.

Phase 1 (preprocessing) — The negation of φ is converted into negation normal form, and $\Diamond$ and $\wedge$ are distributed over $\vee$ as much as possible, by applying the equivalences $\Diamond(\psi \vee \gamma) \equiv \Diamond\psi \vee \Diamond\gamma$ and $\delta \wedge (\psi \vee \gamma) \equiv (\delta \wedge \psi) \vee (\delta \wedge \gamma)$. For each disjunct of the resulting formula $\bigvee \varphi_i'$ a system Sys_i is formed consisting of the single sequent

$\mathbf{i} \Rightarrow \varphi'_i$, where $\mathbf{i}$ is a reserved nominal used to denote the state of evaluation in a model. These are the *initial systems* in the execution.

Phase 2 (elimination) — The algorithm now proceed separately on each initial system, Sys_i, by applying to it the transformation rules listed in table 1. The aim is to eliminate from the system all occurring propositional variables. If this is possible for each system, we proceed to phase 3, else the algorithm report failure and terminates. The rules in table 1 are to be read as rewrite rules, i.e., they replace sequents in systems with new sequents or, in the case of the semantic Ackermann-rule, systems with new systems. Note that each actual elimination of a variable is achieved through an application of the semantic Ackermann-rule while the other rules are used to solve the system for the variable to be eliminated, i.e., to bring the system into the right form for the application of this rule. The applicability of the semantic Ackermann-rule can be determined with the help of Lemma 4 and a suitable modal theorem prover.

We will call the sequents of the form $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$ which are introduced by the $\Diamond$-rule *diamond-link sequents*.

Phase 3 (translation) — This phase is reached only if all systems have been reduced to pure systems, i.e., systems Sys_i with $\mathsf{Form}(\mathsf{Sys}_i)$ a pure formula. Let $\mathsf{Sys}_1, \ldots, \mathsf{Sys}_n$ be these systems. Recalling that φ was the input to the algorithm, we will write $\mathsf{pure}(\varphi)$ for the formula $(\mathsf{Form}(\mathsf{Sys}_1) \vee \cdots \vee \mathsf{Form}(\mathsf{Sys}_n))$. The algorithm now computes and returns, as local frame correspondent for the input formula φ, the formula $\forall\overline{y}\exists x_0 \mathrm{ST}(\neg\mathsf{pure}(\varphi), x_0)$ where $\overline{y}$ is the tuple of all occurring variables corresponding to nominals, but with $y_{\mathbf{i}}$ (corresponding to the designated current state nominal $\mathbf{i}$) left free, since a local correspondent is being computed.

A formula on which SemSQEMA succeeds will be called *SemSQEMA-reducible*, or simply $reducible$.

REMARK 5. — A few remarks are in order:

1) Notice the requirement in the Ackermann-rule that the *conjunction* $B_1 \wedge \cdots \wedge B_m$, rather than the individual sequents be downwards monotone. Since monotonicity as a property is generally not preserved under taking subformulae, this ensures a wider applicability of the rule. We could of course further widen the scope of the rule by requiring only global monotonicity.

2) By replacing the requirement of downward monotonicity in the Ackermann-rule by that of negativity, we obtain the original SQEMA-algorithm.

3) Noting (2) and the relationship between monotonicity and polarity discussed above, it should be clear that all SQEMA-reducible formulae are also SemSQEMA-reducible.

4) By adding further transformation rules facilitating some propositional reasoning (as is done in [CON 06b] and [GAB 06]) the algorithm can be strengthened.

□

The algorithm is best illustrated by an example:

EXAMPLE 6. — Consider the input formula $\varphi := \Diamond\Box p \rightarrow \Diamond(\Diamond\Box\neg p \wedge \Box\Box p)$.

Phase 1 of SemSQEMA negates this formula, produces a negation normal form, distributes $\wedge$ and $\Diamond$ over $\vee$, and produces the single initial system

$$\|\mathbf{i} \Rightarrow \Diamond\Box p \wedge \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p). \tag{1}$$

Phase 2 now proceeds, and by applying the $\wedge$ and $\Diamond$-rules to (1), produces the system

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{j} \Rightarrow \Box p \\ \mathbf{i} \Rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p) \end{array} \right. . \tag{2}$$

An application of the $\Box$-rule transforms (2) into

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \Diamond^{-1}\mathbf{j} \Rightarrow p \\ \mathbf{i} \Rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p) \end{array} \right. . \tag{3}$$

System (3) is now ready for the application of the Ackermann rule, as $\mathbf{i} \rightarrow \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p)$ is downward monotone in p. Indeed, as the reader can check, the consequent of this formula is semantically equivalent to $\Box(\Box\Diamond\top \vee \Diamond\Diamond\neg p)$. The Ackermann-rule is now applied producing the system:

$$\left\| \begin{array}{l} \mathbf{i} \Rightarrow \Diamond\mathbf{j} \\ \mathbf{i} \Rightarrow \Box(\Box\Diamond\Diamond^{-1}\mathbf{j} \vee \Diamond\Diamond\neg\Diamond^{-1}\mathbf{j}) \end{array} \right. . \tag{4}$$

The algorithm proceeds to phase 3, with $\mathsf{pure}(\varphi)$ equal to $(\neg\mathbf{i}\vee\Diamond\mathbf{j})\wedge(\neg\mathbf{i}\vee\Box(\Box\Diamond\Diamond^{-1}\mathbf{j}\vee \Diamond\Diamond\neg\Diamond^{-1}\mathbf{j}))$. Negating $\mathsf{pure}(\varphi)$, translating and simplifying yields the first-order local frame correspondent to φ.

Let us note that the original SQEMA-algorithm would fail on this input formula, since it would not be able to separate the positive and negative occurrences of p (specifically those in the last sequent in (2)) required for the applicability of the syntax-based Ackermann-rule. □

The correctness SemSQEMA can be proved in exactly the the same as that of SQEMA ([CON 06b]), only the soundness of the sematic Ackermann-rule must be justified by an appeal to Lemma 3 rather than to the usual syntax based version of that lemma. Thus, we have:

THEOREM 7 (CORRECTNESS). — *If SemSQEMA succeeds on a formula $\varphi \in \mathcal{L}$ then the first-order formula returned is a local frame correspondent for φ.*

2.3. *The semantic Sahlqvist formulae*

In [CON 06b] it was proven that SQEMA successfully computes first-order frame correspondents for all Sahlqvist [SAH 75] and inductive [GOR 06b] formulae. We now define a generalization of the Sahlqvist class, the members of which are all SemSQEMA-reducible, as will be seen.

DEFINITION 8. — *A boxed atom is a propositional variable prefixed with finitely many (possibly none)* $\Box$*'s. A* downward (upward) monotone block *is a formula which is downward (upward) monotone in all propositional variables occurring in it. A* semantic Sahlqvist antecedent *is a formula built up from* $\top$, $\bot$, *boxed atoms, and downward monotone blocks, using* $\wedge$, $\vee$, *and* $\Diamond$. *A* semantic Sahlqvist implication *is a formula of the form* $\varphi \to \mathsf{UpMon}$ *where* φ *is a semantic Sahlqvist antecedent and* UpMon *is an upward monotone block. A* semantic Sahlqvist formula *is built up from semantic Sahlqvist implications by applying* $\wedge$, $\vee$ *and* $\Box$.

Note that:

– Every (semantic) Sahlqvist implication is semantically equivalent to a negation of a (semantic) Sahlqvist antecedent; consequently, every (semantic) Sahlqvist formula is semantically equivalent to the negation of a (semantic) Sahlqvist antecedent.[2]

– Every Sahlqvist formula is a semantic Sahlqvist formula: to define the latter we have simply replaced in the definition of the former 'negative formulae' with 'downward monotone blocks' and 'positive formulae' with 'upward monotone blocks'.

EXAMPLE 9. — The formula $\Diamond\Box p \to \Diamond(\Diamond\Box\neg p \wedge \Box\Box p)$ from Example 6 is a semantic Sahlqvist implication: recall that $\Diamond(\Diamond\Box\neg p \wedge \Box\Box p)$ is an upward monotone block semantically equivalent to the positive formula $\Diamond(\Diamond\Box\bot \wedge \Box\Box p)$.

Likewise, the formula $\Diamond(\Box(\Box\Diamond q \vee \Diamond\Diamond\neg q) \wedge p) \vee \Box q \to \Diamond(\Diamond\Box\neg p \wedge \Box\Box p)$ is a semantic Sahlqvist implication, as the antecedent is built from the boxed atoms p and $\Box q$ and the downward monotone block $\Box(\Box\Diamond q \vee \Diamond\Diamond\neg q) \equiv_{sem} \Box(\Box\Diamond\top \vee \Diamond\Diamond\neg q)$.
$\Box$

The following lemma will be useful, and can be proved along the same lines as Lemma 5.1. in [CON 06b].

LEMMA 10. — *Let* φ *be a semantic Sahlqvist formula, and* φ' *the formula obtained from* $\neg\varphi$ *by importing the negation over all connectives. Then* φ' *is a semantic Sahlqvist antecedent.*

Note that all semantic Sahlqvist formulae are in $\mathcal{L}$. Let a *generalized monotone block* be an $\mathcal{L}_r^n$-formula which is downward monotone in all occurring propositional variables. The class of *generalized semantic Sahlqvist formulae* is obtained by replacing everywhere in the definition of the semantic Sahlqvist formulae 'monotone block' with 'generalized monotone block'.

LEMMA 11. — *Let* Sys *be a system of* SemSQEMA *sequents of the form* $\mathbf{j} \to \beta$, *with* $\mathbf{j}$ *a nominal and* β *a generalized semantic Sahlqvist antecedent built up without using* $\vee$, *except possibly inside monotone blocks. Let* p *be any propositional*

2. This fact could be used to give a simplified, but semantically equivalent, definition of the class of Sahlqvist formulae, see e.g., [CON 06b].

variable occurring in a boxed atom in Sys. Then Sys can be transformed, using SemSQEMA-transformation rules, into a system Sys′, not containing p, and again with all sequents of the form $\mathbf{j} \rightarrow \beta$.

PROOF. — In Sys all occurrences of p which are not in monotone blocks are in boxed atoms, and these last are at most in the scope of $\wedge$ and $\Diamond$. Thus, by applying the $\wedge$ and $\Diamond$-rules one can transform Sys into a system in which each sequent is of the form either (i) $\mathbf{j} \Rightarrow \beta$ for some nominal $\mathbf{j}$ and some generalized semantic Sahlqvist antecedent β in which p occurs only in monotone blocks, or (ii) $\mathbf{j} \Rightarrow \Box^n p$ for some nominal $\mathbf{j}$ and $n \in \mathbb{N}$. The sequents of type (ii) can all by transformed into the form $(\Diamond^{-1})^n \mathbf{j} \Rightarrow p$ by applying the $\Box$-rule, and then the semantic Ackermann-rule can be applied to eliminate p, yielding a system of the desired shape. (Note that the (generalized) monotone blocks are still downward monotone in the remaining propositional variables after the substitution prescribed by the semantic Ackermann-rule has been done.) ■

THEOREM 12. — *All semantic Sahlqvist-formulae are SemSQEMA-reducible, and hence elementary.*

PROOF. — Let φ be any semantic Sahlqvist-formula, given as input to SemSQEMA. In phase 1 $\neg\varphi$ is transformed into a formula of the from $\bigvee \varphi_i$ by exhaustive distribution of $\wedge$ and $\Diamond$ over $\vee$. Thus, each φ_i is a semantic Sahlqvist antecedent in which all occurrences of $\vee$ are within monotone blocks. For each φ the initial system $\|\mathbf{i} \Rightarrow \varphi_i$ is formed. Since each such initial system is of the from required by Lemma 11, the theorem now follows by induction on the number of propositional variables occurring in each φ_i. ■

We note that this result is presented here mainly in order to demonstrate the scope of applicability of SemSQEMA; otherwise, it can be obtained directly from Sahlqvist's theorem by using the facts that every semantic Sahlqvist-formula is semantically equivalent to a standard Sahlqvist-formula (this follows by the analogue of Lyndon's Theorem for $\mathcal{L}$ proven in [RIJ 97]), and that semantic equivalence preserves both local first-order correspondence and d-persistence of formulae.

3. SemSQEMA and canonicity

Besides the first-order correspondence established by SQEMA, all SQEMA-reducible $\mathcal{L}$-formulae axiomatize complete modal logics. To be precise, for any set Σ of SQEMA-reducible $\mathcal{L}$-formulae, the logic $\mathbf{K} \oplus \Sigma$ is strongly sound and complete with respect to its class of Kripke frames. In [CON 06b] this result was established by showing that all SQEMA-reducible $\mathcal{L}$-formulae are *canonical*, i.e., valid on their canonical frames. In this section we outline, in a modular way, a general framework for proving canonicity for formulae reducible by an algorithm like SemSQEMA. We will show why SemSQEMA fails to fit into this framework.

3.1. *D-persistence and SQEMA*

Here we will attempt to extend the framework for proving canonicity of SQEMA-reducible formulae from [CON 06b] to SemSQEMA.

First, note that, although SemSQEMA-takes input from $\mathcal{L}$, its execution invariably leads us into the richer language $\mathcal{L}_r^n$. To cope with the possible shortage of singletons when interpreting the nominals of the latter language over descriptive frames, we make the following definition:

DEFINITION 13. — *An* augmented model *based on a descriptive frame* $\mathfrak{g} = (W, R, \mathbb{W})$ *is any model* $(\mathfrak{g}, V)$ *such that* V *send propositional variables to members of* $\mathbb{W}$*, as usual, and nominals to arbitrary singletons subsets of* W.

DEFINITION 14. — *Let* $\mathcal{M} = (\mathfrak{f}, V)$ *and* $\mathcal{M}' = (\mathfrak{f}, V')$ *be two models over the same (Kripke or general) frame* $\mathfrak{f}$*, and let* $AT_0 \subseteq AT$*. We say that* $\mathcal{M}$ *and* $\mathcal{M}'$ *are* AT_0-related *if*

1) $V'(p) = V(p)$ *or* $V'(p) = W - V(p)$ *for all propositional variable* $p \in AT_0$*, and*

2) $V'(\mathbf{j}) = V(\mathbf{j})$ *for all nominals* $\mathbf{j} \in AT_0$.

The next definition is intended to capture the type of equivalence which holds between the successive systems of sequents obtained during an execution of SQEMA. As will be illustrated later, we have not been able to guarantee that this is also the case for systems obtained by SemSQEMA.

DEFINITION 15. — *Formulae* $\varphi, \psi \in \mathcal{L}_r^n$ *are* transformation equivalent, *denoted* $\varphi \equiv_{\mathrm{tr}} \psi$*, if, for every model* $\mathcal{M} = (\mathfrak{F}, V)$ *such that* $\mathcal{M} \Vdash \varphi$ *there exists an* $(AT(\varphi) \cap AT(\psi))$*-related model* $\mathcal{M} = (\mathfrak{F}, V')$ *such that* $\mathcal{M}' \Vdash \psi$*, and vice versa.*

Formulae $\varphi, \psi \in \mathcal{L}_r^n$ *are* transformation equivalent over descriptive frames, *denoted* $\varphi \equiv_{\mathrm{tr}}^d \varphi$*, if, for every augmented model* $\mathcal{M} = (\mathfrak{g}, V)$ *based on a descriptive frame* $\mathfrak{g}$*, such that* $\mathcal{M} \Vdash \varphi$ *there exists an* $(AT(\varphi) \cap AT(\psi))$*-related augmented model* $\mathcal{M} = (\mathfrak{g}, V')$ *based on* $\mathfrak{g}$ *such that* $\mathcal{M}' \Vdash \psi$*, and vice versa.*

Let us call any algorithm a *SemSQEMA version* if it is like SemSQEMA in all respects except that it could have a possibly different set of transformation rules. Thus, for example, SQEMA is a SemSQEMA version.

DEFINITION 16. — *A SemSQEMA version* Alg *is* sound on descriptive frames *if for every system of sequents* Sys *and every system* Sys′ *obtained from it by the application* Alg*-transformation rules,* $Form(Sys) \equiv_{tr}^d Form(Sys')$.

Similarly, a SemSQEMA version Alg *is* sound on Kripke frames *if for every system of sequents* Sys *and every system* Sys′ *obtained from it by the application* Alg*-transformation rules,* $Form(Sys) \equiv_{tr} Form(Sys')$.

PROPOSITION 17. — *If a SemSQEMA version* Alg *is sound on descriptive frames and on Kripke frames, then all* Alg*-reducible* $\mathcal{L}$*-formulae are d-persistent.*

PROOF. — Suppose that Alg succeeds on $\varphi \in \mathcal{L}$. Further, for simplicity and without loss of generality, assume that the execution does not branch because of disjunctions. We may make this assumption since any conjunction of d-persistent formulae is d-persistent. Let $\mathsf{Sys}_1, \mathsf{Sys}_2, \ldots, \mathsf{Sys}_m$ be the sequence of systems produced during the execution. Hence Sys_1 is the initial system $\| \mathbf{i} \Rightarrow \neg\varphi$, and Sys_m is the final, pure system and $\mathsf{pure}(\varphi) = \mathsf{Form}(\mathsf{Sys}_m)$.

Let $\mathfrak{g} = (W, R, \mathbb{W})$ be a descriptive frame and $w \in W$. Then $(\mathfrak{g}, w) \Vdash \varphi$ iff there is no augmented valuation V on $\mathfrak{g}$ such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}, V) \Vdash \neg\mathbf{i} \vee \neg\varphi$. But, (since $\mathsf{Form}(\mathsf{Sys}_1) = \neg\mathbf{i} \vee \neg\varphi$, and Alg is sound on descriptive frames) the latter is the case iff there is no augmented valuation V on $\mathfrak{g}$ such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}, V) \Vdash \mathsf{pure}(\varphi)$. Since nominals can range over all singletons, the later is the case iff there is no valuation V on $\mathfrak{g}_\sharp$ such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}_\sharp, V) \Vdash \mathsf{pure}(\varphi)$. By soundness on Kripke frames this, in turn, is the case iff there is no valuation V on $\mathfrak{g}_\sharp$ such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}_\sharp, V) \Vdash \neg\mathbf{i} \vee \neg\varphi$. This is the case iff $(\mathfrak{g}_\sharp, w) \Vdash \varphi$. ■

In [CON 06b] it is shown that the original algorithm SQEMA is sound on both descriptive and Kripke frames, and hence that all SQEMA-reducible formulae are canonical. The main hurdle to be overcome there was to show that a suitable analogue of Ackermann's Lemma holds over descriptive frames. Indeed, the lemma does not generalize to descriptive frames without adaptation, as is shown in example 20 below. However, a restricted version does hold, the formulation of which requires the following definition:

DEFINITION 18. — *A formula $\varphi \in \mathcal{L}_r^n$ is* syntactically closed (open) *if all occurrences of nominals and $\Diamond^{-1}$ in φ are positive (negative), and all occurrences of $\Box^{-1}$ in φ are negative (positive) or, equivalently, when written in negation normal form, φ is positive (negative) in all nominals and contains no occurrences of $\Box^{-1}$ ($\Diamond^{-1}$).*

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

LEMMA 19 (ACKERMANN'S LEMMA FOR DESCRIPTIVE FRAMES,[CON 06B]). — *Suppose $A \in \mathcal{L}_r^n$ is a syntactically closed formula not containing p and $B(p) \in \mathcal{L}_r^n$ is a syntactically open formula which is negative in p. Then*

$$((A \rightarrow p) \wedge B(p)) \equiv_{\mathrm{tr}}^{d} B(A/p).$$

The following example shows that we cannot, in general, lift the requirement in Lemma 19 of syntactic closedness and openness of A and B, respectively.

EXAMPLE 20. — Let $\mathfrak{g} = (W, R, \mathbb{W})$ be the general frame with underlying Kripke frame pictured in figure 1. Note that ω is reflexive while all other points are irreflexive. Further, the only successor of $\omega + 1$ is ω, while the relation in the submodel generated by ω is transitive. Let $\mathbb{W} = \{X_1 \cup X_2 \cup X_3 \mid X_i \in \mathbb{X}_i, i = 1, 2, 3\}$, where $\mathbb{X}_1$ contains all finite (possibly empty) sets of natural numbers, $\mathbb{X}_2$ contains $\emptyset$ and all sets of the form $\{x \in W \mid n \leq x \leq \omega\}$ for all $n \in \omega$, and $\mathbb{X}_3 = \{\emptyset, \{\omega + 1\}\}$. It is not

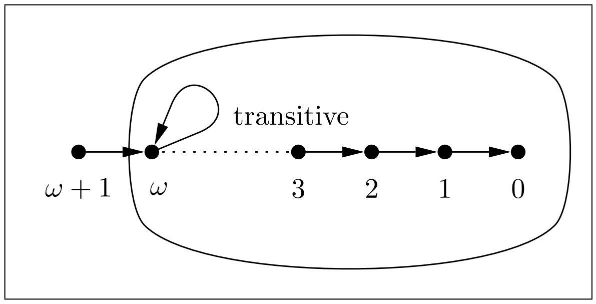

Figure 1. *A descriptive frame*

difficult to check that $\mathfrak{g}$ is descriptive. (This general frame is given in example 8.52 in [CHA 97].)

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

Now, consider a model $\mathcal{M} = (W, R, V)$ based upon the general frame $\mathfrak{g}$ and with $V(\mathbf{i}) = \{\omega+1\}$. Then, consider the formula $\psi := (\Diamond^{-1}\mathbf{i} \to p) \wedge (\neg p \vee \Diamond^{-1}\mathbf{i})$. Ackermann's Lemma (Lemma 3) is applicable to this formula, with $A = \Diamond^{-1}\mathbf{i}$ and $B = (\neg p \vee \Diamond^{-1}\mathbf{i})$. Note that while A is syntactically closed, B is not syntactically open. Now applying the lemma transforms ψ into the tautology $\psi' := \neg\Diamond^{-1}\mathbf{i} \vee \Diamond^{-1}\mathbf{i}$. Even though $\mathcal{M} \Vdash \psi'$, there exists no p-variant $\mathcal{M}'$ (based upon $\mathfrak{g}$) of $\mathcal{M}$ such that $\mathcal{M}' \Vdash \psi$. Indeed, any such p-variant would have to evaluate p to $\{\omega\}$ which is not an admissible set of $\mathfrak{g}$. □

It is easy to prove that whenever the Ackermann-rule is applied during an execution of SQEMA, the syntactic conditions of Lemma 19 are always met. This, however, is *not* the case for SemSQEMA, as is illustrated in example 21 below. It is not known at this stage whether SemSQEMA is sound on descriptive frames or whether SemSQEMA-reducible formulae are always canonical. In sections 5 and 6 below, we introduce two variants of SemSQEMA for which we *are* able to prove soundness on descriptive frames and hence also the canonicity of formulae reducible by them.

EXAMPLE 21. — Consider the formula

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

As the reader can check, SQEMA will fail on this formula. Here is what SemSQEMA does with it. Applying the $\wedge$-rule to the resulting initial system we obtain:

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

Note that this system cannot be solved for p, since neither the positive nor the negative occurrence of p in the first sequent can be isolated by the application of transformation rules. However, the first sequent is downwards monotone in p. Indeed, as we have noted earlier, the formula $\Box\Diamond p \vee \Diamond\Diamond\neg p$ is semantically equivalent to $\Box\Diamond\top \vee \Diamond\Diamond\neg p$. Hence, if we apply the $\Box$-rule to obtain the system

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

we can apply the semantic Ackermann-rule to eliminate p:

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

Solving for q we obtain

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

Now, recalling that $\Box\Diamond q \vee \Diamond\Diamond\neg q \equiv_{sem} \Box\Diamond\top \vee \Diamond\Diamond\neg q$ and applying the semantic Ackermann-rule again we get a pure formula.

Notice, however, that in this application of the semantic Ackermann-rule the antecedent of the first sequent (i.e., the formula A in the Ackermann-equivalence $(A \rightarrow p) \wedge B(p) \equiv B(A)$) is not syntactically closed, since it contains both a negative occurrence of $\Diamond^{-1}$ and a negative nominal occurrence. Attempting to first eliminate the variable q would lead to the same problem. Therefore, we cannot claim d-persistence of the input formula based on its SemSQEMA-reduction.

□

4. Lyndon-type Theorems for syntactically closed and open formulae

In this section we show that syntactically closed upward monotone formulae always have syntactically closed positive equivalents. As a corollary, a similar result holds for syntactically open downward monotone formulae. These theorems are analogues of *Lyndon's monotonicity theorem* for first-order logic ([LYN 59]), and are obtained by making use of a suitable variation of the notion of bisimulation which we will call a *syntactically closed simulation*. These results and techniques will be essential for justifying the canonicity claims we make for the algorithms SemClsSQEMA and SemRepSQEMA presented in sections 5 and 6.

Let $\mathbb{N}^+ := \mathbb{N} \cup \{\infty\}$. As usual $\infty + n = \infty - n = \infty$ for any $n \in \mathbb{N}$.

DEFINITION 22. — *Let $\rho \in \mathbb{N}^+$. A ρ-*bisimulation *relating a pointed model $(\mathcal{M}, m)$ to a pointed model $(\mathcal{N}, n)$ is any family $\{Z_i\}_{0 \leq i < \rho+1}$ of relations $Z_i \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$, between the domains of the models, with $Z_i \subseteq Z_{i+1}$, $0 \leq i < \rho$, satisfying the following conditions:*

(Link) mZ_0n.

(Local harmony) *if* $(u,v) \in \bigcup_{0\leq i<\rho+1} Z_i$ *then* $(\mathcal{M},u) \Vdash a$ *iff* $(\mathcal{N},v) \Vdash a$ *for all* $a \in \mathsf{AT}$.

(Forth) *if* uZ_iv, $i < \rho$, *and* $R^{\mathcal{M}}uu'$, *then* $R^{\mathcal{N}}vv'$ *for some* $v' \in \mathcal{N}$ *such that* $u'Z_{i+1}v'$*; similarly if* uZ_iv *and* $R^{\mathcal{M}}u'u$, *then* $R^{\mathcal{N}}v'v$ *for some* $v' \in \mathcal{N}$ *such that* $u'Z_{i+1}v'$.

(Back) *a similar condition for* uZ_iv, $i < \rho$, *and* $R^{\mathcal{N}}vv'$ / $R^{\mathcal{N}}v'v$.

We will use the notation $(\mathcal{M},m) \rightleftarrows^{\rho} (\mathcal{N},n)$ *to indicate that there exists a* ρ*-bisimulation relating* $(\mathcal{M},m)$ *to* $(\mathcal{N},n)$*. We will write* $Z : (\mathcal{M},m) \rightleftarrows^{\rho} (\mathcal{N},n)$ *if a particular* ρ*-bisimulation* Z *is of importance.*

It should be clear that an ∞-bisimulation is just an ordinary bisimulation for the langauge $\mathcal{L}_r^n$. Recall that the modal depth of an $\mathcal{L}_r^n$-formula φ, denoted $depth(\varphi)$, is the maximum depth of nesting of modal operators in φ. It is well-known that for all $\varphi \in \mathcal{L}_r^n$ with $depth(\varphi) \leq \rho \in \mathbb{N}^+$ it holds that $(\mathcal{M},m) \Vdash \varphi$ iff $(\mathcal{N},n) \Vdash \varphi$, whenever $(\mathcal{M},m) \rightleftarrows^{\rho} (\mathcal{N},n)$.

The following bisimulation notion is designed to preserve syntactically closed $\mathcal{L}_r^n$-formulae which are positive (or upward monotone) in certain propositional variables.

DEFINITION 23. — *Let* $\Theta \subseteq \mathsf{PROP}$ *be a possibly empty set of propositional variables and* $\rho \in \mathbb{N}^+$*. A* syntactically closed Θ-ρ-simulation *relating a pointed model* $(\mathcal{M},m)$ *to a pointed model* $(\mathcal{N},n)$*, is any family* $\{Z_i\}_{0\leq i<\rho+1}$ *of relations* $Z_i \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$*, between the domains of the models, satisfying the following conditions:*

(Link) mZ_0n.

(Asymmetric local harmony for Θ) *If* $(u,v) \in \bigcup Z_i$ *and* $p \in \Theta$*, then* $(\mathcal{M},u) \Vdash p$ *implies* $(\mathcal{N},v) \Vdash p$.

(Asymmetric local harmony for nominals) *if* $(u,v) \in \bigcup Z_i$ *and* $\mathbf{i} \in \mathsf{NOM}$*, then* $(\mathcal{M},u) \Vdash \mathbf{i}$ *implies* $(\mathcal{N},v) \Vdash \mathbf{i}$.

(Local harmony for propositional variables) *if* $(u,v) \in \bigcup Z_i$ *and* $p \in \mathsf{PROP} - \Theta$*, then* $(\mathcal{M},u) \Vdash p$ *iff* $(\mathcal{N},v) \Vdash p$.

(Reversive Forth) *if* uZ_iv, $i < \rho$, *and* $R^{\mathcal{M}}uu'$, *then* $R^{\mathcal{N}}vv'$ *for some* $v' \in \mathcal{N}$ *such that* $u'Z_{i+1}v'$*; similarly if* uZ_iv *and* $R^{\mathcal{M}}u'u$, *then* $R^{\mathcal{N}}v'v$ *for some* $v' \in \mathcal{N}$ *such that* $u'Z_{i+1}v'$.

(Non-Reversive Back) *if* uZ_iv, $i < \rho$, *and* $R^{\mathcal{N}}vv'$, *then* $R^{\mathcal{M}}uu'$ *for some* $u' \in \mathcal{M}$ *such that* $u'Z_{i+1}v'$.

We will use the notation $(\mathcal{M}, m) \rightrightarrows^{\rho}_{SC(\Theta)} (\mathcal{N}, n)$ *to indicate that there exists a syntactically closed* Θ-ρ*-simulation relating* $(\mathcal{M}, m)$ *to* $(\mathcal{N}, n)$*. We will write* $Z : (\mathcal{M}, m) \rightrightarrows^{\rho}_{SC(\Theta)} (\mathcal{N}, n)$ *if a particular* Θ-ρ*-simulation* Z *is of importance.*

LEMMA 24. — *Let* Θ *be a finite, possibly empty set of propositional variables, and* $\rho \in \mathbb{N}^{+}$*. Any syntactically closed* $\mathcal{L}^{n}_{r}$*-formula* φ *of modal depth less that* $\rho + 1$*, which is positive in the propositional variables from* Θ*, is preserved under syntactically closed* Θ-ρ*-simulations.*

PROOF. — By structural induction on φ, written in negation normal form. ■

The next lemma strengthens Lemma 24, by replacing *positivity* with *upward monotonicity*.

LEMMA 25. — *Let* Θ *be a finite, possibly empty set of propositional variables, and* $\rho \in \mathbb{N}^{+}$*. Any syntactically closed* $\mathcal{L}^{n}_{r}$*-formula* φ *of modal depth less than* $\rho + 1$*, which is upwards monotone in the propositional variables from* Θ*, is preserved under syntactically closed* Θ-ρ*-simulations.*

PROOF. — Let φ satisfy the conditions of the lemma and let $\{Z_i\}_{0 \leq i < \rho+1}$ be a syntactically closed Θ-ρ-simulation between the models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$. Suppose that $(\mathcal{M}, m) \Vdash \varphi$.

Let the model $\mathcal{M} \ltimes \mathcal{N} = (W^{\ltimes}, R^{\ltimes}, V^{\ltimes})$ be defined as follows: $W^{\ltimes} = \bigcup Z_i$; $R^{\ltimes}(u, v)(u', v')$ iff $R^{\mathcal{M}}uu'$ and $R^{\mathcal{N}}vv'$; $V^{\ltimes}(p) = \{(u, v) \in \bigcup Z_i \mid u \in V^{\mathcal{M}}(p)\}$ for all propositional variables p; and $V^{\ltimes}(\mathbf{j}) = \{(u, v) \in \bigcup Z_i \mid u \in V^{\mathcal{M}}(\mathbf{j})\}$ for all nominals $\mathbf{j}$. Note that for every nominal $\mathbf{j}$ whose denotation in $\mathcal{M}$ is linked to a point in $\mathcal{N}$ by $\bigcup Z_i$, $V^{\ltimes}(\mathbf{j})$ is a singleton due to the asymmetric local harmony for nominals. All other nominals, however, are interpreted by $V^{\ltimes}$ as $\emptyset$; to remedy this defect we tacitly add to $\mathcal{M} \ltimes \mathcal{N}$ a new point, unrelated to any other by the accessibility relation, where we interpret all those nominals, as well as all propositional variables. The following hold:

(i) $(m, n) \in W^{\ltimes}$, by construction.

(ii) $(\mathcal{M}, m) \rightleftarrows^{\rho} (\mathcal{M} \ltimes \mathcal{N}, (m, n))$, by routine verification that $\{Z'_i\}_{0 \leq i < \rho+1}$ with $Z'_i = \{(u, (u, v)) \mid (u, v) \in Z_i\}$ satisfies definition 22.

(iii) Hence, $(\mathcal{M} \ltimes \mathcal{N}, (m, n)) \Vdash \varphi$.

(iv) Moreover, $(\mathcal{M} \ltimes \mathcal{N}, (m, n)) \rightrightarrows^{\rho}_{SC(\Theta)} (\mathcal{N}, n)$, by routine verification that ${Z''_i}_{0 \leq i < \rho+1}$ with $Z''_i = \{((u, v), v) \mid (u, v) \in Z_i\}$ satisfies definition 23.

Let $\mathcal{M} \leftthreetimes \mathcal{N}$ be obtained from $\mathcal{M} \ltimes \mathcal{N}$ by extending the valuations of the propositional variables in $p \in \Theta$ as follows: for every point in $(u, v) \in \mathcal{M} \leftthreetimes \mathcal{N}$, let $(u, v) \in V^{\leftthreetimes}(p)$ iff $v \in V^{\mathcal{N}}(p)$. Note that $V^{\ltimes}(p) \subseteq V^{\leftthreetimes}(p)$. It follows from the upward monotonicity of φ in the variables from Θ that $(\mathcal{M} \leftthreetimes \mathcal{N}, (m, n)) \Vdash \varphi$. Finally, it is straightforward to check that $(\mathcal{M} \leftthreetimes \mathcal{N}, (m, n)) \rightrightarrows^{\rho}_{SC(\emptyset)} (\mathcal{N}, n)$. Hence, by Lemma 24, $(\mathcal{N}, n) \Vdash \varphi$. ■

Let us denote by $(\mathcal{L}^{n}_{r})^{\rho}_{SC(\Theta)}$ the set of all syntactically closed $\mathcal{L}^{n}_{r}$-formulae, positive in all propositional variables is Θ and of modal depth less that $\rho + 1$. We will write

$(\mathcal{M},m) \Rrightarrow^{\rho}_{SC(\Theta)} (\mathcal{N},n)$ if $(\mathcal{M},m) \Vdash \varphi$ implies $(\mathcal{N},n) \Vdash \varphi$ for all $\varphi \in (\mathcal{L}^n_r)^{\rho}_{SC(\Theta)}$. Note, that $\psi \in (\mathcal{L}^n_r)^{\rho}_{SC(\Theta)}$ iff $\Diamond\psi \in (\mathcal{L}^n_r)^{\rho+1}_{SC(\Theta)}$ iff $\Box\psi \in (\mathcal{L}^n_r)^{\rho+1}_{SC(\Theta)}$ iff $\Diamond^{-1}\psi \in (\mathcal{L}^n_r)^{\rho+1}_{SC(\Theta)}$.

REMARK 26. — $(\mathcal{M},m) \Rrightarrow^{\rho}_{SC(\Theta)} (\mathcal{N},n)$ iff $(\mathcal{N},n) \Vdash \psi$ implies $(\mathcal{M},m) \Vdash \psi$ for all ψ such that $\neg\psi \in (\mathcal{L}^{\rho}_r)^k_{SC(\Theta)}$, i.e., iff all syntactically open formulae, negative in the propositional variables in Θ and of modal depth less than $\rho + 1$ are preserved in passing from $(\mathcal{N},n)$ to $(\mathcal{M},m)$. $\Box$

In what follows we will have to be more precise about the propositional variables and nominals that occur in the language. We will therefore denote by $\mathcal{L}^n_r(\Phi,\Psi)$ the language $\mathcal{L}^n_r$ built over the propositional variables in Φ and the nominals in Ψ. $(\mathcal{L}^n_r(\Phi,\Psi))^{\rho}_{SC(\Theta)}$ is accordingly the restriction of $\mathcal{L}^n_r(\Phi,\Psi)$ to $(\mathcal{L}^n_r)^{\rho}_{SC(\Theta)}$. The relations $\rightrightarrows^{\rho}_{SC(\Theta)(\Phi,\Psi)}$ and $\Rrightarrow^{\rho}_{SC(\Theta)(\Phi,\Psi)}$ are similarly generalized from $\rightrightarrows^{\rho}_{SC(\Theta)}$ and $\Rrightarrow^{\rho}_{SC(\Theta)}$.

LEMMA 27. — *For any pointed models $(\mathcal{M},m)$ and $(\mathcal{N},n)$, set of propositional variables Θ, finite sets Φ and Ψ respectively of propositional variables and nominals, and $k \in \mathbb{N}$,*

$$(\mathcal{M},m) \rightrightarrows^{k}_{SC(\Theta)(\Phi,\Psi)} (\mathcal{N},n) \text{ iff } (\mathcal{M},m) \Rrightarrow^{k}_{SC(\Theta)(\Phi,\Psi)} (\mathcal{N},n).$$

PROOF. — The left-to-right direction is Lemma 24. In the rest of the proof we suppress reference to Φ and Ψ — the only important fact about them is that they are finite, and hence $(\mathcal{L}^n_r)^k_{SC(\Theta)}(\Phi,\Psi)$ is finite, modulo equivalence. We prove the right-to-left direction. Suppose that $(\mathcal{M},m) \Rrightarrow^{k}_{SC(\Theta)} (\mathcal{N},n)$ and let

$$Z_i = \{(u,v) \in W^{\mathcal{M}} \times W^{\mathcal{N}} \mid (\mathcal{M},u) \Rrightarrow^{k-i}_{SC(\Theta)} (\mathcal{N},v)\},$$

for all $0 \leq i \leq k$. We will show that, $\{Z_i\}_{0\leq i\leq k}$ is a syntactically closed Θ-k-simulation linking m and n. By construction, $(m,n) \in Z_0$. It should also be clear that the symmetric and asymmetric local harmony clauses are satisfied by any $(u,v) \in \bigcup_{0\leq i\leq k} Z_i = Z_k$. Suppose that $(u,v) \in Z_i$, for some $0 \leq i < k$, i.e., $(\mathcal{M},u) \Rrightarrow^{k-i}_{SC(\Theta)} (\mathcal{N},v)$. We must show that (u,v) satisfies the back and forth-clauses required by the definition.

To that end, suppose that $R^{\mathcal{M}}uu'$. Let $SC(\Theta)^{k-i-1}(u') = \{\psi \in (\mathcal{L}^n_r)^{k-i-1}_{SC(\Theta)} \mid (\mathcal{M},u') \Vdash \psi\}$, i.e. the set of all $(\mathcal{L}^n_r)^{k-i-1}_{SC(\Theta)}$-formulae true at u'. We may assume that $SC(\Theta)^{k-i-1}(u')$ is finite. Then $\Diamond \bigwedge SC(\Theta)^{k-i-1}(u')$ is an $(\mathcal{L}^n_r)^{k-i}_{SC(\Theta)}$-formula, such that $(\mathcal{M},u) \Vdash \Diamond \bigwedge SC(\Theta)^{k-i-1}(u')$. Hence, $(\mathcal{N},v) \Vdash \Diamond \bigwedge SC(\Theta)^{k-i-1}(u')$, that is to say, v has a $R^{\mathcal{N}}$-successor, say v', such that $(\mathcal{N},v') \Vdash \bigwedge SC(\Theta)^{k-i-1}(u')$. It follows that $(\mathcal{M},u') \Rrightarrow^{k-i-1}_{SC(\Theta)} (\mathcal{N},v')$, and hence that $(u',v') \in Z_{i+1}$. This proves half of the reversive forth-clause. The other half is symmetric, using the formula $\Diamond^{-1} \bigwedge SC(\Theta)^{k-i-1}(u')$ for u' an $R^{\mathcal{M}}$-predecessor of u.

Now for the sake of the non-reversive back-clause, suppose that $(u, v) \in Z_i$ and that $R^{\mathcal{N}}vv'$. Let $SO(\Theta)^{k-i-1}(v') = \{\psi \mid \neg\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^{k-i-1} \;\&\; (\mathcal{N}, v') \Vdash \psi\}$, i.e. the set of all syntactically open formulae of modal depth at most $(k - i - 1)$ and negative in the propositional variables in Θ which are true at v'. Again, we may assume that $SO(\Theta)^{k-i-1}(v')$ is finite. Then $\Diamond \bigwedge SO(\Theta)^{k-i-1}(v')$ is a syntactically open formula of modal depth at most $k - i$, negative in the propositional variables from Θ, and hence, by remark 26, $(\mathcal{M}, u) \Vdash \Diamond \bigwedge SO(\Theta)^{k-i-1}(v')$. Hence, there is a u' such that $R^{\mathcal{M}}uu'$ and $(\mathcal{M}, u') \Vdash \bigwedge SO(\Theta)^{k-i-1}(v')$. Again, by remark 26 it follows that $(\mathcal{M}, u') \Rrightarrow_{SC(\Theta)}^{k-i-1} (\mathcal{N}, v')$ and hence that $(u', v') \in Z_{i+1}$. Note, by the way, that we would not be able to prove a reversive back-clause in a similar way, since $\Diamond^{-1} \bigwedge SO(\Theta)^{k-i-1}(v')$ is not syntactically open. ■

LEMMA 28. — *For any pointed modally saturated (see e.g. [GOR 06a]) models $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$, set of propositional variables Θ, and sets Φ and Ψ respectively of propositional variables and nominals,*

$$(\mathcal{M}, m) \rightrightarrows_{SC(\Theta)(\Phi,\Psi)}^{\infty} (\mathcal{N}, n) \text{ iff } (\mathcal{M}, m) \Rrightarrow_{SC(\Theta)(\Phi,\Psi)}^{\infty} (\mathcal{N}, n).$$

PROOF. — The same as that of Lemma 27, except for the fact that, since we now have to deal with essentially infinite sets of formulae, the modal saturation of $(\mathcal{M}, m)$ and $(\mathcal{N}, n)$ is needed to guarantee the existence of successors satisfying $SC(\Theta)^{k-i-1}(u')$ and $SO(\Theta)^{k-i-1}(v')$. ■

THEOREM 29 (LYNDON-TYPE MONOTONICITY THEOREM FOR SYNTACTICALLY OPEN AND CLOSED FORMULAE). — *A syntactically closed (open) formula $\varphi \in \mathcal{L}_r^n$ is upward (downward) monotone in the propositional variables in a set Θ if and only if it is semantically equivalent to a syntactically closed (open) formula $\varphi' \in \mathcal{L}_r^n$ which is positive (negative) in the propositional variables in Θ and such that $\mathsf{AT}(\varphi') \subseteq \mathsf{AT}(\varphi)$ and $depth(\varphi') \leq depth(\varphi)$.*

PROOF. — The right-to-left direction of the bi-implication is immediate. So, assume that $\varphi \in \mathcal{L}_r^n$ is syntactically closed and upwards monotone in the propositional variables Θ, and suppose that $depth(\varphi) = k$. Let

$$\text{CONS}(\varphi) = \{\psi \in (\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))_{SC(\Theta)}^k \mid \Vdash \varphi \to \psi\}.$$

Note that, because of the bound k on modal depth and the fact that $\mathsf{PROP}(\varphi)$ and $\mathsf{NOM}(\varphi)$ are finite, CONS(φ) is a finite set, modulo semantic equivalence. The proof is complete once we can show that CONS$(\varphi) \Vdash \varphi$, since we can then take φ' to be $\bigwedge$ CONS(φ). To this end, suppose that $(\mathcal{N}, n) \Vdash$ CONS(φ). Let

$$N = \{\psi \mid \neg\psi \in (\mathcal{L}_r^n)_{SC(\Theta)}^k \;\&\; (\mathcal{N}, n) \Vdash \psi\}.$$

Then $N \cup \{\varphi\}$ is satisfiable, for otherwise $N \Vdash \neg\varphi$, i.e $\bigwedge N \Vdash \neg\varphi$. But then $\varphi \Vdash \bigvee\{\neg\psi \mid \psi \in N\}$ and $\bigvee\{\neg\psi \mid \psi \in N\} \in$ CONS(φ) — a contradiction.

Let $(\mathcal{M}, m) \Vdash N \cup \{\varphi\}$. Then, by remark 26, $(\mathcal{M}, m) \Rrightarrow_{SC(\Theta)}^k (\mathcal{N}, n)$, hence, by Lemma 27 we have $(\mathcal{M}, m) \rightrightarrows_{SC(\Theta)}^k (\mathcal{N}, n)$, and then by Lemma 25 it follows that $(\mathcal{N}, n) \Vdash \varphi$. ■

5. SemSQEMA with syntactic restrictions

In this section we introduce and study the modified algorithm SemClsSQEMA. This algorithm is obtained from SemSQEMA by imposing a slight restriction on the applicability of the Ackermann-rule, sufficient to enable us to prove that all SemClsSQEMA-reducible formulae are canonical.

5.1. *The algorithm SemClsSQEMA*

The algorithm SemClsSQEMA is obtained from SemSQEMA by replacing in the latter the semantic Ackermann-rule with the following, restricted version, which we will call the *semantic Ackermann-rule with test for syntactic closedness*:

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

where:

1) p does not occur in $A_1, \ldots, A_n$ or C;

2) All $A_1, \ldots, A_n$ are syntactically closed;

3) $\mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$ is downwards monotone in p.

Thus, this rule is the same as the semantic Ackermann-rule, except for the restriction that $A_1, \ldots, A_n$ are to be syntactically closed. Clearly, we have, as immediate corollary of Theorem 7, the following

THEOREM 30 (CORRECTNESS OF SEMCLSSQEMA). — *If SemClsSQEMA succeeds on a formula $\varphi \in \mathcal{L}$ then the first-order formula returned is a local frame correspondent for φ.*

5.2. *The canonicity of SemClsSQEMA-reducible formulae*

As indicated above, our motivation for restricting SemSQEMA to obtain SemClsSQEMA, is that we can obtain a canonicity result for the formulae reducible by the latter. That canonicity result is to topic of this subsection. We will need some preliminary notions:

We say a model $\mathcal{M} = (W, R, V)$ is *connected* if between every two different points $u, v \in W$ there is a $(R \cup R^{-1})$-path. For the rest of this section we will assume, without loss of generality, that all models are connected, since for the sake of the canonicity proof it is sufficient to work with models generated from the point of evaluation on the reserved nominal $\mathbf{i}$. Formulae $\varphi, \psi \in \mathcal{L}_r^n$ are *cm-equivalent* (for 'equivalent over connected models') if $\mathcal{M} \Vdash \varphi$ iff $\mathcal{M} \Vdash \psi$, for all connected models $\mathcal{M}$.

DEFINITION 31. — *Let $\Sigma \subseteq \mathcal{L}_r^n$ and $\Theta \subseteq$ PROP. We say a formula $\varphi \in \mathcal{L}_r^n$ is* sccm-Θ-reflected between models of Σ *(for 'reflected by syntactically closed Θ-simulations between connected models') if for any two connected models $\mathcal{M} \Vdash \Sigma$ and $\mathcal{N} \Vdash \Sigma$, if $\mathcal{M} \rightrightarrows_{SC(\Theta)}^{\infty} \mathcal{N}$ and $\mathcal{N} \Vdash \varphi$, then $\mathcal{M} \Vdash \varphi$.*

LEMMA 32. — *Any formula $\varphi \in \mathcal{L}_r^n$ which is sccm-$\emptyset$-reflected between models of Σ, and is globally downward monotone in $\Theta \subseteq$ PROP, is cm-equivalent on models of Σ to a syntactically open formula in the vocabulary of φ, which is negative in Θ.*

PROOF. — Let φ be as in the formulation of the lemma. By a construction and argument very similar to those used in the proof of Lemma 25, it can be seen that in fact φ is sccm-Θ-reflected between models of Σ. For the rest of the proof we will write $(\mathcal{L}_r^n)_{SC(\Theta)}$ for $(\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi))_{SC(\Theta)}^{\infty}$. Let

$$\mathrm{CONS}(\varphi) := \{\psi \mid \neg\psi \in (\mathcal{L}_r^n)_{SC(\Theta)} \text{ and } \Sigma \cup \{\varphi\} \Vdash^{mod} \psi\},$$

i.e., $\mathrm{CONS}(\varphi)$ is the set of all syntactically open (global) consequences of φ over models of Σ in the vocabulary of φ which are negative in Θ. If we can show that $\Sigma \cup CONS(\varphi) \Vdash^{mod} \varphi$ we can appeal to compactness and the proof is complete. To that aim, let $\mathcal{M} \Vdash \Sigma \cup CONS(\varphi)$ with $\mathcal{M}$ connected, arbitrarily. Pick any state $m \in \mathcal{M}$ and let $S := \{\gamma \in (\mathcal{L}_r^n)_{SC(\Theta)} \mid (\mathcal{M}, w) \Vdash \gamma\}$. Then there exists a model $\mathcal{N} \Vdash \Sigma \cup \{\varphi\}$ and point $n \in \mathcal{N}$ such that $(\mathcal{N}, n) \Vdash S$, for else there is a finite subset $\{\gamma_1, \ldots, \gamma_n\} \subseteq S$ such that $\neg\gamma_1 \vee \cdots \vee \neg\gamma_n \in \mathrm{CONS}(\varphi)$, which is a contradiction.

Thus, $(\mathcal{M}, m) \Rrightarrow_{SC(\Theta)}^{\infty} (\mathcal{N}, n)$. We may assume, without loss of generality, that both $\mathcal{M}$ and $\mathcal{N}$ are modally saturated, since we know that every model is the bounded morphic image of a modally saturated model (see e.g. [GOR 06a]). Hence by Lemma 28 there exists a syntactically closed simulation $Z : (\mathcal{M}, m) \rightrightarrows_{SC(\Theta)}^{\infty} (\mathcal{N}, n)$. Further, by the definition of a syntactically closed simulation and the assumption that $\mathcal{M}$ is connected, it follows that Z relates every point in $\mathcal{M}$ to some point in $\mathcal{N}$. But then, since φ is sccm-Θ-reflected, we have $\mathcal{M} \Vdash \varphi$. ■

THEOREM 33. — *All SemClsSQEMA-reducible formulae are d-persistent and hence canonical.*

PROOF. — By Lemma 17 it is sufficient to show that SemClsSQEMA is sound both on Kripke and descriptive frames. For that we have to verify that every transformation rule SemClsSQEMA preserves transformation equivalence and transformation equivalence on descriptive frames (definition 15). All cases except one are easy to verify — the only possibly problematic case is to show that the semantic Ackermann rule with test for syntactic closedness preserves transformation equivalence on descriptive frames. Note that for any diamond-link sequent $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$, $\mathsf{Form}(\mathbf{j} \Rightarrow \Diamond\mathbf{k}) = \neg\mathbf{j} \vee \Diamond\mathbf{k}$

is neither syntactically open nor closed. Given a system of sequents Sys, let $Dia(\mathsf{Sys})$ denote the set of all diamond-link sequents appearing in Sys, and $\mathsf{Sys} - Dia(\mathsf{Sys})$ the complement of $Dia(\mathsf{Sys})$ in Sys.

CLAIM 1: Let Sys_i be any system obtained during the (successful or unsuccessful) execution of $\mathsf{SemClsSQEMA}$ on an input formula $\varphi \in \mathcal{L}$. Then the formula $\mathsf{Form}(\mathsf{Sys}_i - Dia(\mathsf{Sys}_i))$ is sccm-$\emptyset$-reflected between all models.

PROOF OF CLAIM 1: We proceed by induction on the application of transformation rules. We can verify the base case by noting that each initial system is of the form $\|\mathbf{i} \Rightarrow \varphi_i$ where $\varphi_i \in \mathcal{L}$. Thus, $\mathsf{Form}(\mathbf{i} \Rightarrow \varphi_i) = \neg\mathbf{i} \vee \varphi_i$ is a syntactically open formula. Therefore, by Lemma 25 and remark 26 $\mathsf{Form}(\mathbf{i} \Rightarrow \varphi_i)$ is sccm-$\emptyset$-reflected between all models.

We may assume, without loss of generality, that all applications of the $\Diamond$-rule and polarity switching rule take place *before* the first application of the Ackermann-rule. Moreover, for all systems Sys obtained before the first application of the Ackerman rule, the formula $\mathsf{Form}(\mathsf{Sys} - Dia(\mathsf{Sys}))$ is syntactically open and hence, as in the base case, $\mathsf{Form}(\mathsf{Sys} - Dia(\mathsf{Sys}))$ satisfies the claim.

Now suppose that Sys_i satisfies the claim. We have to verify that any system Sys'_i obtained from Sys_i by the application of the $\wedge$, $\vee$, $\Box$ or Ackermann-rules satisfies the claim. Since the application of any of the first three of these rules in fact maintains equivalence on models, those cases are immediate. For the case of the Ackermann-rule $\mathsf{Form}(\mathsf{Sys}_i - Dia(\mathsf{Sys}_i))$ must be of the form $(A \to p) \wedge B(p)$ where p does not occur in A, A is syntactically closed, and $B(p)$ is downward monotone in p. (Actually, according to the definition of the Ackermann-rule, there can be an additional conjunct C, not containing p. To keep notation simpler we will, however, write $B(p)$ for $B(p) \wedge C$, as the latter formula will clearly also be downward monotone in p.) It follows for the inductive hypothesis and Lemma 32 that $B(p)$ is cm-equivalent on models of of $(A \to p)$ to a syntactically open formula $B'(p)$ which is negative in p. Now, $\mathsf{Form}(\mathsf{Sys}'_i - Dia(\mathsf{Sys}'_i))$ will be of the from $B(A)$. Suppose that $\mathcal{M}$ and $\mathcal{N}$ are connected models such that $\mathcal{N} \Vdash B(A)$ and that $\mathcal{M} \rightrightarrows^{\infty}_{SC(\emptyset)} \mathcal{N}$. The proof of the claim will be complete if we can show that $\mathcal{M} \Vdash B(A)$. By Ackermann's Lemma (Lemma 3) there is $\mathcal{N}' \sim_p \mathcal{N}$ such that $\mathcal{N}' \vdash (A \to p) \wedge B(p)$, and hence $\mathcal{N}' \vdash (A \to p) \wedge B'(p)$, hence $\mathcal{N}' \vdash B'(A)$. Since $\mathcal{M} \rightrightarrows^{\infty}_{SC(\emptyset)(\mathsf{PROP}-\{p\},\mathsf{NOM})} \mathcal{N}'$, $\mathcal{M}$ is connected, and $B'(A)$ is syntactically open, we have by Lemma 25 that $\mathcal{M} \Vdash B'(A)$. Again by Ackermann's Lemma we can find $\mathcal{M}' \sim_p \mathcal{M}$ such that $\mathcal{M}' \vdash (A \to p) \wedge B'(p)$, hence such that $\mathcal{M}' \vdash (A \to p) \wedge B(p)$, and hence $\mathcal{M}' \vdash B(A)$. But since $\mathcal{M}' \sim_p \mathcal{M}$ and $p \notin \mathsf{PROP}(B(A))$, we have $\mathcal{M} \vdash B(A)$. END PROOF OF CLAIM 1

We can now verify that the semantic Ackermann-rule with test for syntactic closedness preserves transformation equivalence on descriptive frames. For that purpose suppose that $\mathsf{Form}(\mathsf{Sys}_i)$ is of the form $(\neg A \vee p) \wedge B(p) \wedge \mathsf{Form}(Dia(\mathsf{Sys}_i))$ with p not occurring in A, A syntactically closed, and B downward monotone in p. (As before we will take the possible additional conjunct C as part of $B(p)$.) By claim 1, $(\neg A \vee p) \wedge B(p)$ is sccm-$\emptyset$-reflected between all models, and hence $B(p)$ is sccm-

$\emptyset$-reflected between models of $(\neg A \vee p)$. By Lemma 32 $B(p)$ is cm-equivalent on models of $(\neg A \vee p)$ to a syntactically open formula $B'(p)$ which is negative in p. By Lemma 19 $(\neg A \vee p) \wedge B'(p) \equiv_{tr}^{d} B'(A)$. Since $B(A)$ and $B'(A)$ are cm-equivalent on all models and cm-equivalence on all models implies transformation equivalence on descriptive frames, we conclude that $((\neg A \vee p) \wedge B(p)) \equiv_{tr}^{d} B(p)$, and hence $((\neg A \vee p) \wedge B(p)) \wedge \mathsf{Form}(Dia(\mathsf{Sys}_i)) \equiv_{tr}^{d} B(p) \wedge \mathsf{Form}(Dia(\mathsf{Sys}_i))$. ■

Armed with this result, we can now fully extend Sahlqvist's Theorem [SAH 75] to the semantic Sahlqvist formulae, by adding to the correspondence result (Theorem 12) also the accompanying canonicity result:

THEOREM 34. — *All semantic Sahlqvist formulae are canonical.*

PROOF. — By glancing at the proofs of Theorem 12 and Lemma 11, we see that, when SemSQEMA is run on a semantic Sahlqvist formula, the formula $A_1 \vee \cdots \vee A_n$ substituted in the application of the semantic Ackermann-rule is always of the form $(\Diamond^{-1})^{m_1}\mathbf{j}_1 \vee \cdots \vee (\Diamond^{-1})^{m_n}\mathbf{j}_n$. Hence the semantic Ackermann-rule with test for syntactic closedness is if fact applicable. Hence all semantic Sahlqvist formulae are SemClsSQEMA-reducible. The result now follows by Theorem 33. ■

6. SemSQEMA with replacement

In the previous section we introduced and studied the algorithm SemClsSQEMA, obtained from SemSQEMA by imposing a slight restriction on the application of the semantic Ackermann-rule. We were able to show that SemClsSQEMA guarantees the canonicity of formulae reducible by it. This was done by showing that, even though the systems produced by SemClsSQEMA do not always satisfy the syntactic conditions required by Lemma 19, they are in fact always suitably *equivalent* to systems that *do* satisfy those requirements.

In this section we take another approach — we modify SemSQEMA in a way that ensures that systems are always 'syntactically correct', and which requires no restriction on the applicability of the semantic Ackermann-rule. The algorithm we obtain will be called SemRepSQEMA for '*semantic SQEMA with replacement*'.

6.1. *The algorithm SemRepSQEMA*

Theorem 29 guarantees the existence of negative, syntactically open equivalents for formulae which are syntactically open and downward monotone in given propositional variables. As was illustrated in example 21, the syntactic openness of sequents is lost through substitution into formulae which are not negative, effected by the application of the semantic Ackermann rule. If we were thus to replace downward monotone (conjunctions of) sequents with equivalent negative ones *before* we applied the Ackermann-rule, this situation would not arise.

The algorithm SemRepSQEMA is obtained from SemSQEMA by replacing in it the semantic Ackerman-rule with the following *semantic Ackermann-rule with replacement*:

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

where:

1) p does not occur in $A_1, \ldots, A_n$ or C,

2) $\mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$ is downwards monotone in p, and

3) $B'(p)$ is a sequent such that

a) $\mathsf{Form}(B'(p)) \equiv_{\text{sem}} \mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$,

b) $\mathsf{Form}(B'(p))$ is negative in p and syntactically open.

Here is an example of SemRepSQEMA at work:

EXAMPLE 35. — Consider the formula

$$\neg(\Box((\neg q \vee \neg p \vee \Diamond p) \wedge \Diamond\neg r) \wedge \Box(\neg p \vee \Box r) \wedge \Box q \wedge p).$$

SQEMA will fail on this input, as one can check. Let us see if SemRepSQEMA fares any better. After a few applications of the $\wedge$-rule the initial system is transformed into

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

(Strictly speaking, conjunction should be distributed over disjunction on the first sequent but, as this makes no difference to the rest of the execution, we keep the sequent as it is for the sake of compactness of notation.) As the system stands, p cannot be eliminated, but q and r can. Indeed, solving the system for q and r yields

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

which, after two applications of the Ackermann-rule, becomes

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

This is where SQEMA would get stuck. However, the sequent $\mathbf{i} \Rightarrow \Box((\Box^{-1}\neg\mathbf{i} \vee \neg p \vee \Diamond p) \wedge \Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i} \vee \neg p))$ is downward monotone in p, hence SemSQEMA would succeed, but would not prove d-persistence. However, noticing that the formula $(\Box^{-1}\neg\mathbf{i}\vee\neg p\vee\Diamond p)\wedge\Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i}\vee\neg p)$ is semantically equivalent to $\Diamond\Box^{-1}(\Box^{-1}\neg\mathbf{i}\vee \neg p)$, enables us to apply the semantic Ackermann-rule with replacement, yielding

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

□

A simple induction, almost identical to that used to prove Lemma 4.13 in [CON 06b], establishes the following lemma. Recall that diamond-link sequents were sequents of the form $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$ introduced by the $\Diamond$-rule.

LEMMA 36. — *During the entire (successful or unsuccessful) execution of SemRepSQEMA on any $\mathcal{L}$ input formula, all antecedents of non-diamond-link sequents are syntactically closed formulae, while all consequents of non-diamond-link sequents are syntactically open.*

We now have:

THEOREM 37. — *All SemRepSQEMA-reducible $\mathcal{L}$-formulae are locally first-order definable and locally d-persistent.*

PROOF. — The correctness of the algorithm with respect to the first-order equivalents returned follows in the same way as the correctness of SemSQEMA (Theorem 7), as does the soundness on Kripke frames. The soundness on descriptive frames of the transformation rules is also the same as in Theorem 33, except that now the Ackermann-rule with replacement is justified by Lemmas 19 and 36. ■

REMARK 38. — We have proved that both SemClsSQEMA and SemRepSQEMA manage to guarantee canonicity by what may be seen as mitigations of the "semantic thesis". That is to say, both versions reimpose syntactic criteria on the semantic Ackermann-rule which are sufficient to make the Ackermann Lemma for descriptive frames applicable. This lemma requires the formula $\neg A \wedge B(p)$ in the usual Ackermann equivalence to be syntactically open. SemClsSQEMA only requires that A must be syntactically closed (and hence that $\neg A$ is syntactically open), as we are able to show that during the execution B is in fact always suitably equivalent to a syntactically open formula negative in p. On the other hand, SemRepSQEMA ensures that $\neg A \wedge B(p)$ will always be syntactically open with $B(p)$ negative in p, by doing suitable equivalence preserving replacements along the way during the execution. □

To demonstrate the strength of SemRepSQEMA we will show that it succeeds on all *semantically inductive formulae* — the semantic extension of the class of (monadic) inductive formulae introduced in [GOR 06b], defined in the basic modal language $\mathcal{L}$ as follows.

DEFINITION 39. — *Let $\#$ be a symbol not belonging to $\mathcal{L}$. Then a* semantically box-form of $\#$ *in $\mathcal{L}$ is defined recursively as follows:*

1) $\#$ is a semantically box-form of $\#$;

2) If $\mathbf{B}(\#)$ is a semantically box-form of $\#$ then $\Box\mathbf{B}(\#)$ is a semantically box-form of $\#$;

3) If $\mathbf{B}(\#)$ is a semantically box-form of $\#$ and A is an upward monotone formula then $A \to \mathbf{B}(\#)$ is a semantically box-form of $\#$.

Thus, semantically box-forms of $\#$ are, up to semantic equivalence, of the type $\Box(A_1 \to \Box(A_2 \to \ldots \Box(A_n \to \#)\ldots)$, where $A_1, \ldots, A_n$ are upward monotone formulae in $\mathcal{L}$.

DEFINITION 40. — *Given a propositional variable p, a* semantically box-formula of p *is the result $\mathbf{B}(p)$ of substitution of p for $\#$ in any semantically box-form $\mathbf{B}(\#)$. The last occurrence of the variable p is the head of $\mathbf{B}(p)$ and every other occurrence of a variable in $\mathbf{B}(p)$ is inessential there.*

DEFINITION 41. — *A* semantically regular formula *is any modal formula built from upward monotone formulae and negations of semantically box-formulae by applying $\wedge, \vee$, and $\Box$.*

DEFINITION 42. — *The* dependency digraph *of a set $\mathcal{B} = \{\mathbf{B}_1(p_1), \ldots, \mathbf{B}_n(p_n)\}$ of semantically box-formulae is the digraph $G = \langle V, E\rangle$ where $V = \{p_1, \ldots, p_n\}$ is the set of heads in B, and $p_i E p_j$ iff p_i occurs as an inessential variable in a semantically box-formula from $\mathcal{B}$ with a head p_j. A digraph is called* acyclic *if it does not contain oriented cycles.*

DEFINITION 43. — *A* semantically inductive formula *is a semantically regular formula with an acyclic dependency digraph of the set of all semantically box-formulae occurring as subformulae in it.*

We note that semantic Sahlqvist formulae are, up to semantic equivalence, precisely those semantically regular formulae in which the semantically box-formulae are just *boxed atoms*, i.e., propositional variables prefixed by possibly empty strings of boxes. Thus, all semantic Sahlqvist formulae belong to a simple particular case of semantically inductive formulae, where the dependency digraph has no arcs at all.

An example of a semantically inductive formula which is not a semantic Sahlqvist formula is $\neg p \vee \neg\Box(\Diamond(\Diamond\Box\neg p \wedge \Box\Box p) \to \Box q) \vee \Diamond\Box\Box(\Box\Diamond\neg q \vee \Diamond\Diamond q)$.

THEOREM 44. — *All semantically inductive formulae are SemRepSQEMA-reducible, and hence locally first-order definable and locally d-persistent.*

PROOF. — (Sketch) Any semantically inductive formula φ can be regarded as obtained from an inductive formula φ' by replacing some positive subformulae by upwards monotone ones. It has been proved in [CON 06b] that SQEMA succeeds on every inductive formula, by eliminating the variables in an order extending the partial ordering determined by the (acyclic) dependency graph of that formula. Now, given a successful execution of SQEMA on φ', it can be transformed to a successful execution of SemRepSQEMA on φ by applying Theorem 29, whenever necessary, with every application of the semantic Ackermann-rule with replacement. It can be shown by induction on the number of eliminated variables that the conditions of that theorem, guaranteeing the existence of a suitable replacement, are satisfied. ■

Again, this result can be obtained as direct consequence from the local first-order definability and and d-persistence of inductive formulae proved in [GOR 06b]; but here we show that all these formulae are within the range of SemRepSQEMA.

6.2. *Computing syntactically correct equivalents*

The semantic Ackermann-rule with replacement requires negative syntactically open equivalents for downward monotone syntactically open formulae. Theorem 29 guarantees the existence of such equivalents, but if SemRepSQEMA is be called 'algorithm' we need and effective way of computing such equivalents. Strictly speaking, the proof of Theorem 29 does provide a procedure for obtaining such equivalents, albeit a ludicrously inefficient one. Indeed, we just have to note that in that proof, the set $CONS(\varphi)$ is finite, modulo equivalence, and that its members can be effectively computed via an enumeration the members of $(\mathcal{L}_r^n(\mathsf{PROP}(\varphi), \mathsf{NOM}(\varphi)))^k_{SC(\Theta)}$ (again there are finitely many modulo equivalence) and the use of a suitable theorem prover for hybrid logic.

In this section we provide a more efficient algorithm for computing the desired equivalents. This procedure will only work for a special type of monotone formulae (the so-called *separately monotone formulae*), but is often sufficient and will moreover save us the tedious technical detail of the general case.

In many of the examples above, the monotonicity of formulae involved in the application of the Ackermann-rule did not depend on the proper interpretation of the inverse modalities $\Diamond^{-1}$ and $\Box^{-1}$ as inverses of $\Diamond$ and $\Box$. For example, the downwards monotonicity of $\Box^{-1}\neg\mathbf{i} \vee (\Diamond\Box p \wedge \Box\Box\neg p)$ in p can be detected by looking at $\Box_2\neg\mathbf{i} \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$ where $\Box_1$ and $\Box_2$ are two independent modalities. Moreover, the fact that $\mathbf{i}$ is a nominal is also irrelevant: $\Box_2\neg r \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$ is downward monotone in p for any propositional variable r.

With this observation in mind, we introduce the following terminology and definitions. We will refer to the bimodal language with two diamonds $\Diamond_1$ and $\Diamond_2$ as $\mathcal{L}_2$.

DEFINITION 45. — *Given a formula $\varphi \in \mathcal{L}_r^n$, the* separation *of φ, denoted* $\mathsf{Sep}(\varphi)$, *is the $\mathcal{L}_2$-formula obtained by*

1) replacing every occurrence of $\Diamond$ and $\Box$ in φ with $\Diamond_1$ and $\Box_1$, respectively,

2) replacing every occurrence of $\Diamond^{-1}$ and $\Box^{-1}$ in φ with $\Diamond_2$ and $\Box_2$, respectively, and

3) uniformly substituting a fresh propositional variable for every nominal occurring in φ.

For example, $\mathsf{Sep}(\Box^{-1}\neg\mathbf{i} \vee (\Diamond\Box p \wedge \Box\Box\neg p))$ is $\Box_2\neg r \vee (\Diamond_1\Box_1 p \wedge \Box_1\Box_1\neg p)$.

DEFINITION 46. — *A* $\mathcal{L}_r^n$*-formula* φ *is* separately upward monotone *in a propositional variable* p *if* $\mathsf{Sep}(\varphi)$ *is upwards monotone in* p*. The notion of* separate downward monotonicity *is defined similarly.*

Clearly separate monotonicity implies ordinary monotonicity. Recall that the monotonicity and validity (and hence, satisfiability) problems for formulae are interreducible. Now, the validity problem for $\mathcal{L}_r^n$-formulae is EXPTIME-complete ([ARE 00]), while that for $\mathcal{L}_2$ is PSPACE-complete ([HAL 92]). Hence, with the aim of minimizing computational cost, it might be wise to test formula for separate monotonicity first, and only if that fails to test for ordinary monotonicity.

In the rest of this section we present a method for finding negative (positive) syntactically open (closed) equivalents for separately downwards (upwards) monotone formulae. The method will be based on an adaptation of the method of *bisimulation quantifiers*. The idea originates from the '*Pitts quantifiers*' of [PIT 92]. Bisimulation quantifiers have been used to prove uniform interpolation results for the modal μ-calculus in [D'A 02] and for some modal logics in [VIS 96] and [GHI 95]. The normal form used is inspired by that in [CAT 05] and related to that introduced in [JAN 95].

6.2.1. *Disjunctive forms*

If S is a finite (possibly empty) set of $\mathcal{L}_2$-formulae, define ∇S as shorthand for

$$\bigwedge_{\varphi\in S} \Diamond_1\varphi \wedge \Box_1 \bigvee_{\varphi\in S} \varphi,$$

and $\triangle S$ as shorthand for

$$\bigwedge_{\varphi\in S} \Diamond_2\varphi.$$

Note the asymmetry between these definitions — ∇S and $\triangle S$ are defined like this because they will be used to write the separations of syntactically closed formulae. In the case of singleton sets S, we will often write $\nabla\varphi$ and $\triangle\varphi$ for $\nabla\{\varphi\}$ and $\triangle\{\varphi\}$, respectively.

Some standard terminology — *literals* are propositional variables and their negations. For a propositional variable p, the *p-literals* are p and $\neg p$; they are called *complementary literals*. For a set Θ of propositional variables, a Θ-literal is any p-literal for some $p \in \Theta$.

DEFINITION 47. — *The* $\mathcal{L}_2$*-formulae in* disjunctive form *are given recursively by*

$$\varphi ::= \bot \mid \top \mid \chi \wedge \nabla S \wedge \triangle S' \mid \varphi \vee \psi,$$

where χ *is a (possibly empty) conjunction of literals* S *and* S' *are (possibly empty) sets of formulae in disjunctive form. As usual, we identify the empty conjunction with* $\top$*, and the empty disjunction with* $\bot$*. Note that the forms* $\chi \wedge \nabla S$*,* $\nabla S \wedge \triangle S'$ *and* ∇S *can be seen as special cases of* $\chi \wedge \nabla S \wedge \triangle S'$ *with respectively* S'*,* χ*, or both, empty.*

We will call an $\mathcal{L}_2$-formula *syntactically closed* if it contains no positive occurrence of $\Box_2$. (Since we will always be careful to specify in which language we work, this reuse of terminology should cause no confusion. Moreover, in terms of definition 18, all $\mathcal{L}_2$-formulae are syntactically closed, rendering that notion meaningless for such formulae.) Clearly the separation of any syntactically closed $\mathcal{L}_r^n$-formula will be a syntactically closed $\mathcal{L}_2$-formula. Next, we define a translation $(\cdot)^\star$ into disjunctive form of syntactically closed $\mathcal{L}_2$-formulae written in negation normal form. When reading this definition, it is useful to bear the following equivalences in mind: $\nabla\emptyset \equiv_{\mathrm{sem}} \Box_1\bot$, $\nabla\{\varphi, \top\} \equiv_{\mathrm{sem}} \Diamond_1\varphi \wedge \Diamond_1\top \wedge \Box_1(\varphi \vee \top) \equiv_{\mathrm{sem}} \Diamond_1\varphi$, $\nabla\{\top\} \equiv \Diamond_1\top$ and $\varphi \equiv_{\mathrm{sem}} ((\varphi \wedge \Diamond_1\top) \vee (\varphi \wedge \Box_1\bot)$.

$$\begin{aligned}
\top^\star &= \top \\
\bot^\star &= \bot \\
lit^\star &= (lit \wedge \nabla\emptyset) \vee (lit \wedge \nabla\top) \quad \text{for any literal } lit \\
(\varphi \vee \psi)^\star &= \varphi^\star \vee \psi^\star \\
(\Diamond_1\varphi)^\star &= \nabla\{\varphi^\star, \top\} \\
(\Diamond_2\varphi)^\star &= (\nabla\emptyset \wedge \triangle\varphi^\star) \vee (\nabla\top \wedge \triangle\varphi^\star) \\
(\Box_1\varphi)^\star &= \nabla\emptyset \vee \nabla\varphi^\star
\end{aligned}$$

The case for conjunction is more complicated. Consider a formula of the form $\bigwedge S$. If S is such that $S = S' \cup \{\top\}$, $S = S' \cup \{\bot\}$, or $S = S' \cup \{\varphi \vee \psi\}$ we translate as follows

$$\begin{aligned}
(\bigwedge(S' \cup \{\top\}))^\star &= (\bigwedge S')^\star \\
(\bigwedge(S' \cup \{\bot\}))^\star &= \bot \\
(\bigwedge(S' \cup \{\varphi \vee \psi\}))^\star &= (\bigwedge(S' \cup \{\varphi\}))^\star \vee (\bigwedge(S' \cup \{\psi\}))^\star
\end{aligned}$$

Note that in the last case above we are in effect distributing the conjunction over the disjunction. If S does not contain $\top$, $\bot$, or a disjunction, it means that every formula in S is either a literal or a formula of the form $\Diamond_1\psi$, $\Box_1\psi$, or $\Diamond_2\psi$. We now define the following sets:

$$\begin{aligned}
S_{\Diamond_1} &= \{\psi \mid \Diamond_1\psi \in S\} \\
S_{\Box_1} &= \{\psi \mid \Box_1\psi \in S\} \\
S_{\Diamond_2} &= \{\psi \mid \Diamond_2\psi \in S\}
\end{aligned}$$

Lastly, let S_{lit} be the subset of all literals in S. If $S_{\Diamond_1} \neq \emptyset$, then the intuition is that any point satisfying $\bigwedge S$ must satisfy each member of S_{lit}, every member of $S_{\Diamond_2}$ must be satisfied at some R_2-successor, and every member of $S_{\Diamond_1}$ must be satisfied at some R_1-successor which also satisfies all members of $S_{\Box_1}$. We must also take into account the fact that there may be R_1-successors not satisfying any member of $S_{\diamond_1}$, but which still have to satisfy all members of $S_{\Box_1}$. Hence, if $S_{\diamond_1} \neq \emptyset$, we translate thus:

$$(\bigwedge S)^\star = \bigwedge S_{lit} \wedge \nabla\{(\varphi \wedge \bigwedge S_{\Box_1})^\star \mid \varphi \in S_{\diamond_1} \cup \{\top\}\} \wedge \triangle\{\psi^\star \mid \psi \in S_{\diamond_2}\}.$$

If, on the other hand, $S_{\diamond_1} = \emptyset$, points satisfying the formula can either have no R_1-successors, or have R_1-successors, each satisfying every member of $S_{\Box_1}$. Hence, if $S_{\diamond_1} = \emptyset$, let

$$\begin{aligned}(\bigwedge S)^\star \quad = \quad & (\bigwedge S_{lit} \wedge \nabla\emptyset \wedge \triangle\{\psi^\star \mid \psi \in S_{\diamond_2}\}) \\ & \vee(\bigwedge S_{lit} \wedge \nabla\{(\bigwedge S_{\Box_1})^\star\} \wedge \triangle\{\psi^\star \mid \psi \in S_{\diamond_2}\}).\end{aligned}$$

It should be clear that $\varphi \equiv_{\mathrm{sem}} \varphi^\star$ for every syntactically closed $\mathcal{L}_2$-formula φ in negation normal form. Here is an example:

EXAMPLE 48. — Consider the formula $r \wedge \diamond_1(\diamond_1\Box_1\neg p \wedge \Box_1\Box_1 p \wedge \neg q)$. Since it contains no occurrences of $\diamond_2$, we will omit the subscripts and simply write $\diamond$ and $\Box$ for $\diamond_1$ and $\Box_1$, respectively. It is translated into disjunctive form as follows:

$$\begin{aligned}& (r \wedge \diamond(\diamond\Box\neg p \wedge \Box\Box p \wedge \neg q))^\star \\ =& r \wedge \nabla\{(\diamond\Box\neg p \wedge \Box\Box p \wedge \neg q)^\star, \top\} \\ =& r \wedge \nabla\{\neg q \wedge \nabla\{(\Box\neg p \wedge \Box p)^\star, (\Box p)^\star\}, \top\} \\ =& r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\}\end{aligned}$$

□

6.2.2. *Simulation quantifiers and biased simulations*

Via disjunctive forms and the following definition we will transform upward monotone syntactically closed formulae into positive ones.

DEFINITION 49. — *Let φ be an $\mathcal{L}_2$-formula in disjunctive form and $\overline{p}$ a vector of propositional variables. We define $\exists^+\overline{p}.\varphi$ inductively as follows:*

$$\begin{aligned}\exists^+\overline{p}.\bot \quad &= \quad \bot \\ \exists^+\overline{p}.\top \quad &= \quad \top \\ \exists^+\overline{p}.(\chi \wedge \nabla S \wedge \triangle S') \quad &= \quad \chi' \wedge \nabla\{\exists^+\overline{p}.\psi \mid \psi \in S\} \wedge \triangle\{\exists^+\overline{p}.\psi \mid \psi \in S'\} \\ \exists^+\overline{p}.(\varphi \vee \psi) \quad &= \quad \exists^+\overline{p}.\varphi \vee \exists^+\overline{p}.\psi\end{aligned}$$

where χ' is $\bot$ when χ is inconsistent (i.e. when χ contains complementary literals), or otherwise, if χ is consistent, χ' is obtained from χ by removing (by simply deleting) all occurrences of negative $\overline{p}$-literals. $\exists^+\overline{p}$ is called a simulation quantifier.

Note that $\exists^+\overline{p}.\varphi$ is positive in all variables in $\overline{p}$. We want to show that $\exists^+p.\varphi \equiv_{\mathrm{sem}} \varphi$ for all formulae φ that are upward monotone in p. To that aim the following definition, which is essentially a separated version of a syntactically closed Θ-simulation (definition 23).

DEFINITION 50. — *Let* $\mathcal{M} = (W^{\mathcal{M}}, R_1^{\mathcal{M}}, R_2^{\mathcal{M}}, V^{\mathcal{M}})$ *and* $\mathcal{N} = (W^{\mathcal{N}}, R_1^{\mathcal{N}}, R_2^{\mathcal{N}}, V^{\mathcal{N}})$ *be* $\mathcal{L}_2$*-models. Let* Θ *be a set of propositional variables. A* Θ-biased simulation *between* $\mathcal{M}$ *and* $\mathcal{N}$ *is a nonempty binary relation* $Z \subseteq W^{\mathcal{M}} \times W^{\mathcal{N}}$ *satisfying, for all* $(u, v) \in W^{\mathcal{M}} \times W^{\mathcal{N}}$ *such that* uZv*, the following conditions:*

(local harmony) $(\mathcal{M}, u) \Vdash p$ *iff* $(\mathcal{N}, v) \Vdash p$ *for all propositional variables* $p \notin \Theta$*,*

(asymmetric local harmony) $(\mathcal{M}, u) \Vdash p$ *only if* $(\mathcal{N}, v) \Vdash p$*, for all propositional variables* $p \in \Theta$*,*

(symmetric forth) *if* $R_1^{\mathcal{M}}uu'$ *(respectively,* $R_2^{\mathcal{M}}uu'$*) then there exists a point* $v' \in W^{\mathcal{N}}$ *such that* $u'Zv'$ *and* $R_1^{\mathcal{N}}vv'$ *(respectively,* $R_2^{\mathcal{N}}vv'$*), and*

(asymmetric back) *if* $R_1^{\mathcal{N}}vv'$ *then there exists a point* $u' \in W^{\mathcal{M}}$ *such that* $u'Zv'$ *and* $R_1^{\mathcal{N}}uu'$*.*

We will write $\mathcal{M} \hookrightarrow_\Theta \mathcal{N}$ *if there exists a* Θ*-biased simulation between models* $\mathcal{M}$ *and* $\mathcal{N}$*, or* $(\mathcal{M}, m) \hookrightarrow_\Theta (\mathcal{N}, n)$ *if there is a* Θ*-biased simulation linking* m *and* n*.*

A straightforward adaptation of the proof of Lemma 25 establishes the next lemma.

LEMMA 51. — *For all* $\mathcal{L}_2$*-models* $(\mathcal{M}, m)$ *and* $(\mathcal{N}, n)$ *such that* $(\mathcal{M}, m) \hookrightarrow_\Theta (\mathcal{N}, n)$*, and all syntactically closed* $\mathcal{L}_2$*-formulae* φ*, which are upward monotone in the variables in* Θ*, it holds that* $(\mathcal{M}, m) \Vdash \varphi$ *only if* $(\mathcal{N}, n) \Vdash \varphi$*.*

LEMMA 52. — *Let* $\varphi \in \mathcal{L}_2$ *be a syntactically closed formula in disjunctive from and* $\overline{p}$ *a vector of propositional variables. Then,* $(\mathcal{M}, m) \Vdash \varphi$ *implies* $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$*.*

PROOF. — By induction on φ. ■

The next theorem motivates why we call $\exists^+\overline{p}$ a 'simulation quantifier':

PROPOSITION 53. — *Let* $\varphi \in \mathcal{L}_2$ *be a syntactically closed formula in disjunctive from. Then, for any model* $(\mathcal{N}, n)$ *and any vector of propositional variables* $\overline{p}$*,*

$$(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$$

if and only if there exists a model $(\mathcal{M}, m)$ *such that*

$$(\mathcal{M}, m) \Vdash \varphi \quad \textit{and} \quad (\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n).$$

PROOF. — We proceed by induction on φ. The base case for $\top$ is trivial, as is the inductive step for φ of the form $\psi_1 \vee \psi_2$. We consider the case for φ of the form $\chi \wedge \nabla S \wedge \triangle S'$.

The bottom-to-top direction is easy. By Lemma 52, $(\mathcal{M}, m) \Vdash \varphi$ implies $(\mathcal{M}, m) \Vdash \exists^+\overline{p}.\varphi$. Also note that $\exists^+\overline{p}.\varphi$ is positive in all propositional variables in $\overline{p}$. We can now appeal to Lemma 51, and conclude that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$.

Conversely, suppose that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$. By the inductive hypothesis it follows that, for each pair (ψ, s) such that $\psi \in S$, $R_1^{\mathcal{N}}ns$ and $(\mathcal{N}, s) \Vdash \exists^+\overline{p}.\psi$, there exists a pointed model $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)})$ such that $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \Vdash \psi$ and $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$. Moreover, since every $R_1^{\mathcal{N}}$-successor s of n satisfies $\exists^+\overline{p}.\psi$ for some formula $\psi \in S$, we have that $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$ with $(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \Vdash \psi$ for some $\psi \in S$.

Also by the inductive hypothesis, for every $\psi \in S'$ there exists a point $s \in W^{\mathcal{N}}$ and a pointed model $(\mathcal{M}_\psi, m_\psi)$ such that $R_2^{\mathcal{N}}ns$, $(\mathcal{M}_\psi, m_\psi) \hookrightarrow_{\overline{p}} (\mathcal{N}, s)$ and $(\mathcal{M}_\psi, m_\psi) \Vdash \psi$.

Now we construct the desired model $(\mathcal{M}, m)$ by first taking the disjoint union of the models in the sets

$$\left\{(\mathcal{M}_{(\psi,s)}, m_{(\psi,s)}) \mid \psi \in S, R_1^{\mathcal{N}}ns, (\mathcal{N}, s) \Vdash \exists^+\overline{p}.\psi\right\}$$

and

$$\{(\mathcal{M}_\psi, m_\psi) \mid \psi \in S'\}.$$

To this disjoint union we add a new point m and make it an R_1-predecessor of each $m_{(\psi,s)}$, and an R_2-predecessor of each m_ψ. To complete the model we make all propositional variables occurring positively in χ true at m while all other propositional variables are declared false there. By construction $(\mathcal{M}, m) \Vdash \varphi$ and $(\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n)$. ■

THEOREM 54. — *Let $\varphi \in \mathcal{L}_2$ be a syntactically closed formula in disjunctive from which is upward monotone in $\overline{p}$. Then $\varphi \equiv_{\text{sem}} \exists^+\overline{p}.\varphi$.*

PROOF. — As remarked before, $\Vdash \varphi \rightarrow \exists^+\overline{p}.\varphi$. Conversely, suppose that $(\mathcal{N}, n) \Vdash \exists^+\overline{p}.\varphi$. By proposition 53 there exists a model $(\mathcal{M}, m)$ such that $(\mathcal{M}, m) \Vdash \varphi$ and $(\mathcal{M}, m) \hookrightarrow_{\overline{p}} (\mathcal{N}, n)$. But, by Lemma 51, φ is preserved under $\overline{p}$-biased simulations, i.e., $(\mathcal{N}, n) \Vdash \varphi$. ■

Theorem 54 gives us a procedure to compute positive equivalents for upward monotone syntactically closed $\mathcal{L}_2$-formulae, written in disjunctive form. This is easily converted into a procedure for computing negative equivalents for separately downward monotone syntactically open $\mathcal{L}_r^n$-formulae. To be precise, suppose that $\varphi \in \mathcal{L}_r^n$ is syntactically open and separately downward monotone in the propositional variable p. We compute the desired equivalent of φ as follows:

1. Negation: Negate φ and apply the usual procedure to rewrite the $\neg\varphi$ in negation normal form, obtaining φ'. The formula φ' is syntactically closed and separately upward monotone in p.

2. Separation: Separate φ' by calculation $\mathsf{Sep}(\varphi')$. The formula $\mathsf{Sep}(\varphi')$ will be a syntactically closed $\mathcal{L}_2$-formula which is upward monotone in p.

3. Disjunctive form: Transform $\mathsf{Sep}(\varphi')$ into disjunctive form by applying the translation $(\cdot)^\star$, i.e., by calculating $(\mathsf{Sep}(\varphi'))^\star$.

4. Elimination of negative p-occurrences: Calculate $\exists^+ p.(\mathsf{Sep}(\varphi'))^\star$. This formula is positive in p.

5. Obtaining positive $\mathcal{L}_r^n$-equivalent: Reverse step 3 as far as possible by applying the inverse of the translation function $(\cdot)^\star$ and the definitions of ∇S and $\triangle S'$. Lastly, obtain an $\mathcal{L}_r^n$-formula by applying the inverse of Sep.

6. Second negation: Negate the resulting formula again to obtain a syntactically open formula, negative in p, and semantically equivalent to φ.

Let us illustrate this procedure with an example.

EXAMPLE 55. — In example 21 we used the fact that the formula $\gamma = \neg\mathbf{i} \vee \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q)$ was downward monotone in p. Indeed, it is even separately downward monotone in p, as $\mathsf{Sep}(\gamma) = \neg r \vee \Box(\Box\Diamond p \vee \Diamond\Diamond\neg p \vee q)$ is downward monotone in p. (Since there are no inverse modalities involved in this formula, we can omit the subscripts in the separated from without risk of confusion.) Let us compute a negative equivalent for this formula using the method of simulation quantifiers, described above. Negating and rewriting in negation normal form we obtain $r \wedge \Diamond(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q)$. In example 48 this formula was translated into disjunctive form, thus:

$$\begin{aligned} & (r \wedge \Diamond(\Diamond\Box\neg p \wedge \Box\Box p \wedge \neg q))^\star \\ = \quad & r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \end{aligned}$$

Next, application of the simulation quantifier $\exists^+ p$ yields

$$\begin{aligned} & \exists^+ p.(r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\neg p \wedge p\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\}) \\ = \quad & r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\bot\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \end{aligned}$$

Reversing the $(\cdot)^\star$-translation step by step yields

$$\begin{aligned} & r \wedge \nabla\{\neg q \wedge \nabla\{\nabla\emptyset \vee \nabla\{\bot\}, \nabla\emptyset \vee \nabla\{p\}\}, \top\} \\ = \quad & r \wedge \nabla\{\neg q \wedge \nabla\{\Box\bot, \Box p\}, \top\} \\ = \quad & r \wedge \nabla\{\neg q \wedge \Diamond\Box\bot \wedge \Diamond\Box p \wedge \Box(\Box\bot \vee \Box p), \top\} \\ = \quad & r \wedge \Diamond(\neg q \wedge \Diamond\Box\bot \wedge \Diamond\Box p \wedge \Box(\Box\bot \vee \Box p)) \end{aligned}$$

Lastly, undoing the Sep-function and negating yields a syntactically open equivalent, negative in p:

$$\neg\mathbf{i} \vee \Box(q \vee \Box\Diamond\top \vee \Box\Diamond\neg p \vee \Diamond(\Diamond\top \wedge \Diamond\neg p))$$

Admittedly, this equivalent could be simpler. Indeed, as noted in example 21, it is in fact equivalent to $\neg\mathbf{i} \vee \Box(\Box\Diamond\top \vee \Diamond\Diamond\neg p \vee q)$. The introduction of the subformula $\Box\Diamond\neg p$ is worrying, as this quantifier pattern is often the cause of SQEMA's failure. However, for the input formula in example 21 this causes no problem, as the reader can check. More sophisticated strategies for undoing $(\cdot)^\star$ should be able to minimize this problem since all that we are doing at the moment is applying the definition in reverse. □

EXAMPLE 56. — In example 35 the monotonicity of the formula

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

was used in an application of the semantic Ackermann rule with replacement. As this formula is not separately monotone, the method presented in this section will not suffice to compute an equivalent negative in p in this case. Indeed,

$$(\Box_2\neg t \vee \neg p \vee \Diamond_1 p) \wedge \Diamond_1\Box_2(\Box_2\neg t \vee \neg p)$$

is not monotone in p. □

In summary, monotone sequents in SemRepSQEMA-executions often satisfy the stronger property of separated monotonicity. Syntactically correct equivalents of these sequents can be computed by using the method of simulation quantifiers presented in this section. However, as example 56 illustrates, SemRepSQEMA-executions may give rise to sequents which are monotone but not separately monotone. To compute syntactically correct equivalents for these, stronger methods will have to be considered.

7. Conclusion

In this paper we explored the application of the modal monotonicity-based version of Ackermann's Lemma to the computation of first-order frame equivalents for modal formulae and to proving their canonicity. This was done through appropriate modifications of the algorithm SQEMA. Specifically, we introduced three extensions of SQEMA which employ the monotonicity-based (semantic) version of the Ackermann-rule. Two of these extensions guarantee the canonicity of the formulae on which they succeed, at the expense of either a restricted scope applicability (in the case of SemClsSQEMA) or of a possibly dramatic increase in the complexity (in the case of SemRepSQEMA). One of the most important open questions related to the present study, is whether SemSQEMA, being the most efficient and general of the three extensions of SQEMA proposed here, guarantees canonicity, too.

In any case, a natural question arising here is to estimate the complexities of each of the proposed extensions of SQEMA. We have not investigated this question in any depth, as we do not expect the worst case complexities to be of good practical value here, while computing the average case complexities would be a computational challenge going beyond the scope and purpose of this paper.

Furthermore, while SemSQEMA seems the version easiest to apply, it does not necessarily have optimal scope of applicability. The reason for this is that direct application of the semantic Ackermann-rule can produce syntactically bad-shaped sequents which could impede the successful termination of the algorithm. Thus, canonicity considerations apart, the idea of replacing certain formulae by syntactically more suitable ones before or during the applications of the Ackermann rule, remains quite relevant for the sake of improving the chances of eventual success. However, the questions of what replacements are eventually useful and how to compute them, still remain largely unexplored.

Finally, we have not yet reached the limits of the semantic approach to computing first-order frame equivalents (and proving canonicity) of modal formulae. In its pure form this approach calls for a gradual elimination of any syntactic transformations in favour of effectively executable semantic tests — possibly much more expensive computationally, but further extending the scope of applicability of the method. Going further along that way, however, we are bound to face the ubiquitous tradeoff between the generality and efficiency of any algorithmic approach. We believe that the present paper offers a good equilibrium between these.

Acknowledgements

We wish to thank Dimiter Vakarelov for discussions and advice on previous versions of this paper, as well as the anonymous referee for careful reading and useful comments. The second author gratefully acknowledges the financial support of his work by the National Research Foundation of South Africa through an ICT research grant.

8. References

[ACK 35] ACKERMANN W., "Untersuchung über das Eliminationsproblem der mathematischen Logic", *Mathematische Annalen*, vol. 10, 1935, p. 390-413.

[ARE 00] ARECES C., BLACKBURN P., MARX M., "The Computational Complexity of Hybrid Temporal Logics", *Logic Journal of the IGPL*, vol. 8(5), 2000, p. 653-679.

[BLA 01] BLACKBURN P., DE RIJKE M., VENEMA Y., *Modal Logic*, Cambridge University Press, 2001.

[CAT 05] TEN CATE B., "A note on the length of explicit definitions in modal logic", Unpublished note, 2005.

[CHA 91] CHAGROVA L. A., "An undecidable problem in correspondence theory", *JSL*, vol. 56, 1991, p. 1261-1272.

[CHA 97] CHAGROV A., ZAKHARYASCHEV M., *Modal Logic*, Oxford, 1997.

[CON 06a] CONRADIE W., "On the strength and scope of DLS", *Journal of Applied Non-Classical Logics*, vol. 16(3-4), 2006, p. 279–296.

[CON 06b] CONRADIE W., GORANKO V., VAKARELOV D., "Algorithmic correspondence and completeness in modal logic I : The core algorithm SQEMA", *Logical Methods in Computer Science*, vol. 2(1:5), 2006.

[CON 06c] CONRADIE W., GORANKO V., VAKARELOV D., "Algorithmic correspondence and completeness in modal logic II. Polyadic and hybrid extensions of the algorithm SQEMA", *Journal of Logic and Computation*, vol. 16, 2006, p. 579-612.

[D'A 02] D'AGOSTINO G., LENZI G., "On modal μ-calculus with explicit interpolants", report num. PP-2002-17, 2002, Institute for Logic, Language and Computaion, University of Amsterdam.

[GAB 92] GABBAY D. M., OHLBACH H.-J., "Quantifier elimination in second-order predicate logic", *South African Computer Journal*, vol. 7, 1992, p. 35-43.

[GAB 06] GABBAY D. M., SCHMIDT R., SZALAS A., *Second-Order Quantifier Elimination: Mathematical Foundations, Computational Aspects and Applications*, College Publications, 2006, to appear.

[GHI 95] GHILARDI S., "An algebraic theory of normal forms", *Annals Pure and Applied Logic*, vol. 71, num. 3, 1995, p. 189–245.

[GOR 03] GORANKO V., HUSTADT U., SCHMIDT R. A., VAKARELOV D., "SCAN Is Complete for All Sahlqvist Formulae", BERGHAMMER R., MÖLLER B., STRUTH G., Eds., *RelMiCS*, vol. 3051 of *Lecture Notes in Computer Science*, Springer, 2003, p. 149-162.

[GOR 06a] GORANKO V., OTTO M., "*Model theory of modal logic*", p. 249–329, Elsevier, 2006, In [**?**].

[GOR 06b] GORANKO V., VAKARELOV D., "Elementary Canonical Formulae: Extending Sahlqvist Theorem", *Annals of Pure and Applied Logic*, vol. 141(1-2), 2006, p. 180-217.

[HAL 92] HALPERN J. Y., MOSES Y. O., "A guide to completeness and complexity for modal logics of knowledge and belief", *Artificial Intelligence*, vol. 54, 1992, p. 319-379.

[JAN 95] JANIN D., WALUKIEWICZ I., "Automata for the modal μ-calculus and related results", *Mathematical foundations of computer science 1995 (Prague)*, vol. 969 of *Lecture Notes in Computer Science*, p. 552–562, Springer, Berlin, 1995.

[LYN 59] LYNDON R. C., "Properties preserved under homomorphism", *Pacific Journal of Mathematics*, vol. 9, 1959, p. 143-154.

[PIT 92] PITTS A. M., "On an Interpretation of Second Order Quantification in First Order Intuitionistic Propositional Logic", *Journal of Symbolic Logic*, vol. 57, num. 1, 1992, p. 33-52.

[RIJ 97] DE RIJKE M., KURTONINA N., "Simulating without Negation", *Journal of Logic and Computation*, vol. 7, 1997, p. 1-22.

[SAH 75] SAHLQVIST H., "Correspondence and completeness in the first and second-order semantics for modal logic", KANGER S., Ed., *Proc. of the 3rd Scandinavian Logic Symposium, Uppsala 1973*, Amsterdam, 1975, Springer-Verlag, p. 110-143.

[SZA 93] SZALAS A., "On the correspondence between modal and classical logic: An automated approach", *Journal of Logic and Computation*, vol. 3, 1993, p. 605–620.

[VIS 96] VISSER A., "Uniform interpolation and layered bisimulation", *Gödel '96 (Brno, 1996)*, vol. 6 of *Lecture Notes Logic*, p. 139–164, Springer, Berlin, 1996.