Algorithmic correspondence and completeness in modal logic
V. Recursive extensions of SQEMA.

Willem Conradie

Department of Mathematics, University of Johannesburg, South-Africa

Valentin Goranko

Department of Informatics and Mathematical Modelling, Technical University of Denmark

Dimitar Vakarelov

Faculty of Mathematics and Computer Science, Sofia University, Bulgaria

Abstract

The previously introduced algorithm SQEMA computes first-order frame equivalents for modal formulae and also proves their canonicity. Here we extend SQEMA with an additional rule based on a recursive version of Ackermann's lemma, which enables the algorithm to compute local frame equivalents of modal formulae in the extension of first-order logic with monadic least fixed-points FOμ. This computation operates by transforming input formulae into locally frame equivalent ones in the pure fragment of the hybrid mu-calculus. In particular, we prove that the recursive extension of SQEMA succeeds on the class of 'recursive formulae'. We also show that a certain version of this algorithm guarantees the canonicity of the formulae on which it succeeds.

Keywords: Modal logic, correspondence theory, canonicity, first-order logic with fixed-points, SQEMA.

1 Introduction

Since the introduction of Kripke semantics in the early 60s and van Benthem's standard translation [31] of modal formulae into universal monadic second-order (UMSO) formulae over Kripke frames in the mid 1970s, modal languages have come to be seen as special fragments of UMSO logic with respect to frame validity.

Particular interest has developed since then in *correspondence theory* between modal formulae and first-order (or elementary) formulae with respect to definability of frame properties, classical accounts of which can be found in [31] and [32]. Some of the highlights of that theory are the discoveries of the class of first-order definable and canonical Sahlqvist formulae ([26]) on the one hand, and of some notable cases of non first-order definability of important modal principles, such as the axioms of

Gödel-Löb, McKinsey, and Segerberg, on the other hand. Via the standard frame translation of modal formulae into UMSO, correspondence theory can be seen as an instance of the second-order quantifier elimination problem, an up to date account of which can be found in [16].

The part of correspondence theory dealing with first-order definability and canonicity of modal formulae has recently been advanced by the extension of the class of Sahlqvist formulae to the class of inductive formulae in [20], [19], [21] and the development of the algorithm SQEMA in [9], [10] for computing first-order equivalents and proving canonicity of modal formulae.

Meanwhile, it has been observed in several recent publications including [24], [21], [34], [33], [18], [30] that many naturally arising non-elementary modal formulae, such as Gödel-Löb axiom GL and Segerberg's induction axiom IND, define frame conditions which are expressible in the extension of first-order logic with fixed-points of monadic predicates FOμ — an important and well-behaved extension of FO, with better understood semantics and model theory than the full (universal) MSO [14], [23]. Thus, the trend to develop correspondence theory between modal logic and FOμ emerges naturally. Some important recent contributions towards such correspondence theory include:

▷ [24], where a recursive version of Ackermann's lemma [1] was proved, and it was shown how FOμ-equivalents of monadic second-order formulae may be obtained through applications of that lemma. In particular, second-order translations of various modal formulae, such as GL, can be reduced to FOμ in this way. To bring formulae into the form which makes the lemma applicable, rules in the style of the DLS-algorithm ([13]) are applied. More recently, an extension DLS* of the DLS-algorithm was implemented, which computes FOμ-equivalents of monadic second-order formulae.

▷ [33], where the class of 'PIA' ('positive antecedent implies atom') first-order formulae was identified as precisely those (up to logical equivalence) first-order formulae which have the 'Intersection Property' with respect to a given predicate letter P, meaning that the set of interpretations of P satisfying the formula in a given structure is closed under intersection, and consequently there is a minimal such interpretation of P. A PIA formula $\phi(P(\mathbf{x}), \mathbf{Q}) \to P(\mathbf{x})$ essentially says that P is a pre-fixed-point of the operator $\phi(P(\mathbf{x}), \mathbf{Q})$, and therefore the minimal interpretation of a predicate satisfying a PIA formula is a least fixed-point; in fact, the extension of FO with iterated minimization over PIA-formulae in that extension is shown in [33] to be expressively equivalent to FOμ. The theme of modal correspondence in FOμ is carried further in [34] where a number of examples are considered and a general result, similar to Theorem 3.5 in the present paper, is obtained.

▷ In [21] the class of so called 'regular polyadic modal formulae' [1] was introduced. It was shown there that they all have local equivalents in FOμ. The regular formulae include almost all well-known examples of FOμ-definable modal formulae.

The present paper develops a direct algorithmic method for computing FOμ-

[1] Here, these formulae will be called more appropriately 'recursive'.

equivalents of modal formulae, and as such it is intended as a further contribution towards the building of a correspondence theory between modal logic and FOμ. This algorithmic method is based on extensions of the algorithm SQEMA, obtained by employing a recursive version of the modal Ackermann lemma. It is similar, but different, from the adaptation of Ackermann's lemma used in [24]. In particular, we show that the recursive extensions of SQEMA developed here are powerful enough to compute FOμ-correspondents of all monadic regular formulae, mentioned above.

2 Preliminaries

Apart from the background presented in this section, we assume the reader's familiarity with the basic notions pertaining to the semantics and model theory of modal logic (see e.g., [3] or [18]), the modal μ-calculus (see e.g., [6]), and with the core algorithm SQEMA ([9]), described for the reader's convenience in the Appendix.

2.1 First-order logic with monadic fixed-points

Here we will only consider the extension of first-order logic with *monadic* fixed-points, i.e., where all predicates P to which the μ-operator is applied are unary. Given a first-order language FO of any fixed signature τ, we define its extension FO$\mu(\tau)$ with monadic least fixed-point operators by adding to the inductive definition of FO-formulae the clause:

(MLFP) If $\varphi(P, x, \mathbf{Q}, \mathbf{y})$ is a formula positive in P, with free variables amongst x and the tuple $\mathbf{y}$, (monadic) predicate symbols amongst P and the tuple $\mathbf{Q}$, then $\mu(P, x).\varphi(P, x, \mathbf{Q}, \mathbf{y})[u, \mathbf{y}]$ is a formula.

The semantics of the formula $\mu(P, x).\varphi(P, x, \mathbf{Q}, \mathbf{y})[u, \mathbf{y}]$ is given by the least fixed-point of the monotone set-operator defined by the formula φ [14], [23]. The dual, greatest fixed-point operator, is defined as expected:

$$\nu(P, x).\varphi(P, x, \mathbf{Q}, \mathbf{y})[u, \mathbf{y}] := \neg\mu(P, x).\neg\varphi(\neg P/P, x, \mathbf{Q}, \mathbf{y})[u, \mathbf{y}]$$

While FOμ is a rather expressive extension of FO, it still shares some nice properties with FO, e.g., the downward Löwenheim-Skolem theorem [15] and the 0-1 law (see [5]). For further background on FOμ see [14] and [23].

2.2 Hybrid modal logics and hybrid mu-calculus

Given a monadic multi-modal language ML(σ) of some modal signature σ, we associate with it a first-order language FO(σ) with a range of unary predicates corresponding to the propositional variables, and a range of binary relations corresponding to the modal operators in ML(σ). By way of the well-known standard translation ST (see e.g., [31], [3], [18])) every modal formula φ from ML(σ) is translated to a FO(σ)-formula ST(φ, x) of one free variable x. Hereafter, we will assume a monadic multi-modal signature σ fixed and will omit it in the notation for the various associated languages. We also consider the following extensions of ML:

- The hybrid extension HML with nominals (denoted by $\mathbf{i}, \mathbf{j}, \mathbf{k}, \ldots$), the universal modality, and inverse modalities for all modal operators in the basic signature σ.

The fragment of HML without the universal modality will be denoted by HML^l ('HML local').

The standard translation for ML readily extends over HML by adding the obvious clauses for the extra modalities and a clause for the nominals: $\mathrm{ST}(\mathbf{i}, x) := (x = y_{\mathbf{i}})$ where $y_{\mathbf{i}}$ is a reserved variable associated with the nominal $\mathbf{i}$.

The *pure fragment of HML*, denoted PHML, consists of all HML-formulae containing no occurrences of propositional variables. PHML^l, the pure fragment of HML^l, is defined likewise.

- The modal μ-calculus, MLμ, is obtained by adding a clause constructing the least fixed-point $\mu p.\varphi(p, \mathbf{q})$ [2] for every MLμ-formula $\varphi(p, \mathbf{q})$ positive in the variable p.

  The standard translation for ML extends over MLμ into FOμ, by adding a clause for the μ-operator:

  $$\mathrm{ST}(\mu p.\varphi(p, \mathbf{q}), x) := \mu(P, z).\mathrm{ST}(\varphi(p, \mathbf{q}), z)[x],$$

  where P is the unary predicate symbol in FOμ corresponding to the propositional variable p.

- The *hybrid μ-calculus* language HMLμ is obtained by merging the languages HML and MLμ; its standard translation to FOμ combines the standard translations of these sublanguages. The fragment of HMLμ without the universal modality will be denoted by $\mathrm{HML}^l\mu$.

  The *pure fragment of HMLμ*, denoted PHMLμ, consists of all HMLμ-formulae where all occurrences of propositional variables are bound by μ-operators. The fragment of PHMLμ without the universal modality will be denoted by $\mathrm{PHML}^l\mu$.

The standard translation $\mathrm{ST}(\varphi, x)$ of a modal formula from any of the modal languages above relates the local truth of φ (at the state denoted by x) to the local truth of $\mathrm{ST}(\varphi)$ *in Kripke models*. By universally quantifying $\mathrm{ST}(\varphi, x)$ over all unary predicate symbols occurring in $\mathrm{ST}(\varphi, x)$ and not bound by fixed-point operators, we obtain an MSO-formula $\overline{\mathrm{ST}(\varphi, x)}$ that expresses the local validity (at the state denoted by x of the Kripke frame) of φ.

The hybrid μ-calculus HMLμ was studied in [27], where it is proved that the satisfiability problem for this logic is decidable in ExpTime. This result is important for the purposes of the present paper, since the execution of the recursive SQEMA-algorithm, introduced further, produces HMLμ-formulae which we need to formally reason about. In fact, the output language of the recursive SQEMA-algorithm will be the pure fragment PHMLμ of HMLμ.

3 Modal correspondence to FOμ

A formula φ from any of the modal languages introduced above is said to be *locally equivalent* to a FOμ-formula $\theta(x)$ if for every Kripke frame $\mathfrak{F}$ for the respective signature and state $w \in \mathfrak{F}$, we have that $\mathfrak{F}, w \Vdash \varphi$ iff $\mathfrak{F} \models \theta[x := w]$. Similarly, φ is *globally equivalent* to a FOμ-sentence θ if, for every Kripke frame $\mathfrak{F}$, we have that $\mathfrak{F} \Vdash \varphi$ iff $\mathfrak{F} \models \theta$. A modal formula is *locally (globally) FOμ-definable* if it is locally

[2] We use the same symbol, viz. μ, to denote least fixed-point operators in both FOμ and MLμ.

(globally) equivalent to some FOμ-formula (FOμ-sentence). Clearly, local implies global FOμ-definability, but the converse is false. Indeed, the formula $\Box\Diamond\Box\Box p \rightarrow \Diamond\Diamond\Box\Diamond p$ has $\forall x \exists y Rxy$ as global frame correspondent. However, in [31] van Benthem shows that this formula violates the Löwenheim-Skolem theorem, which, as was mentioned above, holds for FOμ.

As already noted, some non-elementary modal formulae have (local) equivalents in FOμ, while others do not (see examples in subsection 3.3). Thus, the problem of recognizing the modal formulae that are (locally) definable in FOμ arises [3]. No explicit syntactic or model-theoretic criteria seem to be known as yet, and we conjecture that FOμ-definability of ML(σ)-formulae is undecidable.

Here we will develop semi-algorithms which effectively compute FOμ-equivalents for a fairly large class of modal formulae. In what follows we will work with (arbitrary) monadic multi-modal logics, but generalizations to polyadic modal languages will be discussed briefly in the concluding section.

3.1 Recursive modal formulae

A large class of polyadic modal formulae was effectively defined in [20] under the name 'regular formulae'. Here we will consider the subclass of monadic multimodal regular formulae, explicitly defined first in [18], from where we import the definitions below. For reasons that will become clear, hereafter we will call these formulae 'recursive'.

Definition 3.1 Let an arbitrary monadic multi-modal language ML(σ) be fixed and let # be a symbol not belonging to ML(σ). Then a *box-form of* # in ML(σ) is defined recursively as follows:

(i) # is a box-form of #;

(ii) If $\mathbf{B}(\#)$ is a box-form of # and $\Box$ is a box-modality in ML(σ) then $\Box\mathbf{B}(\#)$ is a box-form of #;

(iii) If $\mathbf{B}(\#)$ is a box-form of # and A is a positive ML(σ)-formula then $A \rightarrow \mathbf{B}(\#)$ is a box-form of #.

Thus, box-forms of # are, up to semantic equivalence, of the type $\Box_1(A_1 \rightarrow \Box_2(A_2 \rightarrow \ldots \Box_n(A_n \rightarrow \#)\ldots)$, where $\Box_1, \ldots, \Box_n$ are (possibly empty) strings of box-modalities and $A_1, \ldots, A_n$ are positive formulae in ML(σ).

Definition 3.2 Given a propositional variable p, a *box-formula of* p is the result $\mathbf{B}(p)$ of substitution of p for # in any box-form $\mathbf{B}(\#)$. The rightmost occurrence of the variable p is the *head* of $\mathbf{B}(p)$.

Definition 3.3 A *(monadic) recursive formula* in ML(σ) is any ML(σ)-formula built from positive formulae and negations of box-formulae by applying conjunctions, disjunctions, and boxes.

Example 3.4 Here are some examples, showing that, although the definition might seem somewhat involved, recursive formulae are quite common.

[3] The converse problem, of which FOμ-formulae are modally definable on Kripke frames, although not less interesting, will not be discussed here.

- The Gödel-Löb formula $\Box(\Box p \to p) \to \Box p$ can be equivalently rewritten as a recursive formula $\neg\Box(\Box p \to p) \vee \Box p$.
- The Sambin-Boolos formula $\Box(\Box p \leftrightarrow p) \to \Box p$ is an incomplete weakening of the Gödel-Löb formula. It is equivalent to the recursive formula $\neg\Box(\Box p \to p) \vee \neg\Box(p \to \Box p) \vee \Box p$.
- Segerberg's bimodal induction axiom $IND = [2](q \to [1]q) \to (q \to [2]q)$ is equivalent to the recursive formula $\neg[2](q \to [1]q) \vee \neg q \vee [2]q$.
- van Benthem's 'Cyclic return' formula from [33]: $(\Diamond p \wedge \Box(p \to \Box p)) \to p$ is not a recursive formula, but becomes semantically equivalent to one after a polarity change, i.e., substitution of $\neg p$ for p:
  $(\Diamond\neg p \wedge \Box(\neg p \to \Box\neg p)) \to \neg p \equiv \Box p \vee \neg\Box(\Diamond p \to p) \vee \neg p$.

More examples will be considered later.

Theorem 3.5 ([21]) *Every recursive formula has a local correspondent in FOμ, which can be obtained effectively.*

The key observation for the result above is that all recursive formulae have minimal valuations that can be used to eliminate the propositional variables, which are recursively defined and eventually expressible in FOμ.

We note that Theorem 3.5 subsumes and extends Theorem 4 in [34] and, essentially, Theorem 8 in [33], because the syntax of recursive formulae allow for a deeper nesting of PIA subformulae.

Furthermore, the class of recursive formulae contains all conjunctions of inductive formulae studied in [20], [8], [21], which in turn strictly subsume all Sahlqvist formulae, so Theorem 3.5 can be regarded as an extension of the definability part of the Sahlqvist theorem. However, we see from the examples above that it cannot match the canonicity part of it, because there are Kripke incomplete recursive formulae. On the other hand, we believe that every modal formula which has a minimal valuation expressible in FOμ is semantically equivalent to a recursive formula.

3.2 Definability of the pure hybrid modal mu-calculus in FOμ

Since the standard translation of any PHMLμ-formula in FOμ does not contain free predicate symbols, the proof of the following proposition, which will be used further in Section 5.3, is immediate.

Proposition 3.6 *Every PHMLμ-formula has a local equivalent in FOμ.*

As we will see further in some examples, a PHMLμ-formula can turn out to be FO-definable if the successive unfoldings of the fixed-point operators stabilize at a finite stage. If that unfolding is computed within PHMLμ, we can test for stabilization at each stage in ExpTime, using the decidability of satisfiability in HMLμ [27]. This procedure yields explicit definitions of such fixed-points in PHML and hence PHML-equivalents for the PHMLμ-formulae involved.

It is, of course, also possible for a PHMLμ-formula to have an equivalent in PHML despite its not having finitely-stabilizing fixed-points. Since the execution of the recursive version of SQEMA (to be introduced further) applied to input formulae from HML^l remains within $\mathrm{HML}^l\mu$, it suffices to look for FO-equivalents of $\mathrm{HML}^l\mu$-formulae. The question of whether a $\mathrm{HML}^l\mu$-formula has an equivalent in HML^l was proved decidable in [12] via an adaptation of an argument in [25].

3.3 On non-definability of modal formulae in FOμ

Not all modal formulae have (local) equivalents in FOμ. For instance, McKinsey's formula $\mathrm{McK} = \Box\Diamond p \to \Diamond\Box p$ does not have such an equivalent. Indeed, as noted in [18], the proof of the Downward Löwenheim-Skolem theorem for FOμ in [15] actually produces a countable elementary submodel of any infinite model satisfying a given FOμ-sentence. On the other hand, van Benthem has constructed in [31] an uncountable frame in which McK is valid, while it is not valid in any countable subframe of it. This argument should likewise apply to other modal reduction principles which are not FO-definable.

Another method for obtaining examples of non FOμ-definable modal formulae, partly noted in [33], too, is based on the following: it was proved in [5] that all FOμ-formulae satisfy the so called 'transfer theorem' stating that asymptotically almost sure validity in finite frames implies validity in the countable random frame (also known as the Rado graph), and consequently they satisfy the Zero-One law. On the other hand, both the Zero-One law [22] and the transfer property [17] fail in modal logic. The simplest known example of a modal formula which cannot be FOμ-definable because of failing the transfer property is $\neg\Box\Box(p \leftrightarrow \Diamond\neg p)$; for a more general description of such formulae see [17].

4 Recursive extensions of SQEMA

In this section we introduce extensions of the algorithm SQEMA that compute local FOμ-equivalents of monadic multi-modal formulae.[4] After giving examples of the execution of these extensions, we will prove their correctness with respect to local FOμ-equivalence, and also the completeness of the basic extension $\mathsf{SQEMA}^{\mathsf{rec}}$ with respect to the class of recursive formulae. The exposition will assume familiarity with the basic SQEMA-algorithm, the details of which are recalled in the appendix for the reader's convenience.

4.1 Recursive Modal Ackermann Lemma

Given a tuple of propositional variables $\mathbf{p}$, we say that two models $\mathcal{M}$ and $\mathcal{M}'$ are $\mathbf{p}$-*variants* of each other, denoted $\mathcal{M} \sim_{\mathbf{p}} \mathcal{M}'$, if they are based on the same frame and differ from each other at most in the valuation of the variables in $\mathbf{p}$.

Consider a list of operators $A_1(X_1, \ldots, X_n), \ldots, A_n(X_1, \ldots, X_n)$, where each $A_i : (\mathcal{P}(W))^n \to \mathcal{P}(W)$, for $i = 1, \ldots, n$, is monotone in all arguments. Then the vector operator $\langle A_1, \ldots, A_n\rangle : (\mathcal{P}(W))^n \to (\mathcal{P}(W))^n$ given by

$$\langle A_1, \ldots, A_n\rangle(X_1, \ldots, X_n) \mapsto \langle A_1(X_1, \ldots, X_n), \ldots, A_n(X_1, \ldots, X_n)\rangle$$

[4] Polyadic extensions are briefly discussed in Section 6.

has a least fixed-point $\mu\langle X_1,\ldots,X_n\rangle.\langle A_1(X_1,\ldots,X_n),\ldots,A_1(X_1,\ldots,X_n)\rangle$, which by Bekič's lemma (see e.g., [2]) can be computed coordinate-wise, too.

The following is a recursive extension of the modal version of Ackermann's lemma [9].

Lemma 4.1 (Recursive Ackermann Lemma) *Let $\mathbf{p}=(p_1,\ldots,p_n)$ and let $A_1(\mathbf{p}),\ldots,A_n(\mathbf{p}),B(\mathbf{p})$ be HMLμ-formulae, with $A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})$ positive and $B(\mathbf{p})$ negative in each of the variables $p_1,\ldots,p_n$. Let also $B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})\rangle/\mathbf{p}]$ be the result of the component-wise substitution of the tuple $\mu\mathbf{p}.\langle A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})\rangle$ for $\mathbf{p}$ in B. Then for every Kripke model $\mathcal{M}$ the following holds:*

$$\mathcal{M}\Vdash B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})\rangle/\mathbf{p}]$$
iff there is a model $\mathcal{M}'\sim_{\mathbf{p}}\mathcal{M}$ such that
$$\mathcal{M}'\Vdash(A_1(\mathbf{p})\to p_1)\wedge\cdots\wedge(A_n(\mathbf{p})\to p_n)\wedge B(\mathbf{p}).$$

Proof. To avoid the technical overhead, we will do the proof for the case $n=2$; the proof of the general case is a relatively straightforward generalization. To simplify the argument a little we will use Bekič's lemma [2], according to which $\mu\langle p_1,p_2\rangle.\langle A_1(p_1,p_2),A_2(p_1,p_2)\rangle=\langle\mu p_1.A_1(p_1,\mu p_2.A_2(p_1,p_2)),\mu p_2.A_2(\mu p_1.A_1(p_1,p_2),p_2)\rangle$.

First, let $\mathcal{M}=(W,R,V)\Vdash B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),A_2(\mathbf{p})\rangle/\mathbf{p}]$ and let us put $V'(p_1)=V(\mu p_1.A_1(p_1,\mu p_2.A_2(p_1,p_2)))$, $V'(p_2)=V(\mu p_2.A_2(\mu p_1.A_1(p_1,p_2),p_2))$.

Then $V'(B(\mathbf{p}))=V(B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),A_2(\mathbf{p})\rangle/\mathbf{p}])$, hence $\mathcal{M}'=(W,R,V')\Vdash B(\mathbf{p})$.

Moreover, $\mathcal{M}'\Vdash A_1(p_1,p_2)\to p_1$, since $V'(A_1(p_1,p_2))=V(A_1(\mu p_1.A_1(p_1,\mu p_2.A_2(p_1,p_2)),\mu p_2.A_2(\mu p_1.A_1(p_1,p_2),p_2)))=V(\mu p_1.A_1(p_1,\mu p_2.A_2(p_1,p_2)))=V'(p_1)$.

Likewise, $\mathcal{M}'\Vdash A_2(p_1,p_2)\to p_2$.

Conversely, let for some $\mathbf{p}$-variant $\mathcal{M}'=(W,R,V')$ of $\mathcal{M}$:

$$\mathcal{M}'\Vdash(A_1(p_1,p_2)\to p_1)\wedge(A_2(p_1,p_2)\to p_2)\wedge B(p_1,p_2).$$

Then $\langle V'(p_1),V'(p_2)\rangle$ is a pre-fixed-point of $\langle A_1(p_1,p_2),A_2(p_1,p_2)\rangle$ in $\mathcal{M}'$, hence $V'(\mu p_1.A_1(p_1,\mu p_2.A_2(p_1,p_2)))\subseteq V'(p_1)$ and $V'(\mu p_2.A_2(\mu p_1.A_1(p_1,p_2),p_2))\subseteq V'(p_2)$ since $\mu\langle p_1,p_2\rangle.\langle A_1(p_1,p_2),A_2(p_1,p_2)\rangle$ is the least among all such pre-fixed-points.

Hence, by the downward monotonicity of $B(\mathbf{p})$, $\mathcal{M}'\Vdash B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),A_2(\mathbf{p})\rangle/\mathbf{p}]$.

Since $\mathcal{M}\sim_{\mathbf{p}}\mathcal{M}'$ and p_1,p_2 do not occur free in $B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),A_2(\mathbf{p})\rangle/\mathbf{p}]$, we then have $\mathcal{M}\Vdash B[\mu\mathbf{p}.\langle A_1(\mathbf{p}),A_2(\mathbf{p})\rangle/\mathbf{p}]$. □

Another recursive version of Ackermann Lemma and some generalizations formulated in an algebraic form was introduced in [29] and [30].

4.2 Recursive versions of the Ackermann Rule

Based upon Lemma 4.1 we can now formulate the following transformation rule:

Recursive Ackermann rule (RAR):

$$\left\| \begin{array}{l} A_1(p_1,\ldots,p_n) \Rightarrow p_1, \\ \ldots \\ A_n(p_1,\ldots,p_n) \Rightarrow p_n \\ B_1(p_1,\ldots,p_n) \\ \ldots \\ B_m(p_1,\ldots,p_n) \\ C_1 \\ \ldots \\ C_k \end{array} \right. \quad \text{is replaced with} \quad \left\| \begin{array}{l} B_1[\mu\mathbf{p}.\langle A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})\rangle/\mathbf{p}], \\ \ldots \\ B_m[\mu\mathbf{p}.\langle A_1(\mathbf{p}),\ldots,A_n(\mathbf{p})\rangle/\mathbf{p}], \\ C_1 \\ \ldots \\ C_k \end{array} \right.$$

where:

- $p_1,\ldots,p_n$ are different variables and $\mathbf{p} = (p_1,\ldots,p_n)$.
- $A_1,\ldots,A_n$ are positive formulae in each of $p_1,\ldots,p_n$,
- $B_1,\ldots,B_m$ are negative formulae in each of $p_1,\ldots,p_n$, and
- $C_1,\ldots,C_k$ are formulae not containing any of $p_1,\ldots,p_n$.

The variables $p_1,\ldots,p_n$ above can be assumed different, since sequents $A' \Rightarrow p$ and $A'' \Rightarrow p$ can be combined into one $A' \vee A'' \Rightarrow p$. Likewise, for the purposes of the rule, all formulae $C_1,\ldots,C_k$ can be assumed to be combined conjunctively into one.

Note that the rule RAR acts on the entire system. The simplest, and most commonly used form of the rule RAR, involves only one fixed-point. All examples given further will only involve this case.

4.3 Versions of the Recursive SQEMA

By adding the recursive Ackerman-rule (RAR) one can obtain different recursive extensions of the basic SQEMA-algorithm. These versions differ in their strategies regarding the computation of the fixed-point operators introduced by RAR.

4.3.1 The standard recursive SQEMA

The Recursive SQEMA, denoted $\mathsf{SQEMA}^{\mathsf{rec}}$, is obtained from SQEMA (see Appendix) by adding RAR as a transformation rule. If phase 2 of $\mathsf{SQEMA}^{\mathsf{rec}}$ completes successfully, then the formula $\mathsf{pure}(\phi)$ is in PHMLμ. Accordingly, the translation step of $\mathsf{SQEMA}^{\mathsf{rec}}$ yields a formula in FOμ, and no attempt is made to reduce this result to FO by eliminating the fix point operators. Of course, it is also possible, depending on the user's preference, to obtain an output in PHMLμ by terminating the algorithm before the translation is done. We note that $\mathsf{SQEMA}^{\mathsf{rec}}$ can just as well work on any HMLμ-formulae as input.

4.3.2 The Eager Recursive SQEMA

This version of $\mathsf{SQEMA}^{\mathsf{rec}}$ works as follows. At every application of the recursive Ackermann Rule, the algorithm runs a procedure attempting to compute the least fixed-point $\mu p.A(p)$ substituted by the rule, by computing the successive iterations *symbolically*, starting with $\bot$, within the hybrid modal μ-calculus HMLμ to which all formulae of the current SQEMA-system belong, and testing at every stage if the computation has stabilized, by testing an equivalence of the type $A^n(\bot) \equiv A^{n+1}(\bot)$ – which is decidable in ExpTime [27]. If the equivalence holds at some stage, the fixed-point is computed and it is a formula in HML, which the Ackermann Rule substitutes instead of the $\mu p.A(p)$.

We call this version of SQEMA$^{\mathsf{rec}}$ the Recursive SQEMA with eager computation of the fixed-points, or the Eager Recursive SQEMA for short, denoted SQEMA$^{\mathsf{eagrec}}$. It can be used to compute FO-equivalents of modal formulae in some cases when the basic SQEMA fails, thus extending and strengthening the correspondence part of the basic SQEMA. In fact the canonicity results for SQEMA also extend to SQEMA$^{\mathsf{eagrec}}$, as will be shown in Section 5.2.

4.4 Examples

We now present some examples illustrating the execution of SQEMA$^{\mathsf{rec}}$ and SQEMA$^{\mathsf{eagrec}}$ on different input formulae.

Example 4.2 [Gödel-Löb formula] Given the Gödel-Löb formula $\Box(\Box p \to p) \to \Box p$ as an input, the first phase of the algorithm preprocesses it to $\Box(\Diamond\neg p \vee p) \wedge \Diamond\neg p$.

In phase 2 one initial system is produced, namely

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

By applying the $\wedge$ and then the $\Box$-rule, this is transformed into

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

Applying the left-shift-$\vee$-rule transforms (b) in (c). The RAR is now applicable, yielding (d).

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

Note that SQEMA would get stuck at system (c), as the standard Ackermann rule cannot be applied since the positive and negative occurrences of p in the first sequent cannot be separated.

Translating and negating (phase 3), we get (with a slight abuse of notation)

$$\exists x_0(x_0 = i \wedge \forall y(Rx_0y \to \mu(P,x).(Rx_0x \wedge \forall z(Rxz \to P(z)))[y])).$$

Simplifying, we eventually obtain the local equivalent:

$$\forall y(Riy \to \mu(P,x).(Rix \wedge \forall z(Rxz \to P(z)))[y]).$$

This is the semantic condition locally corresponding to the Gödel-Löb formula, viz. local transitivity plus inverse well-foundedness. Since this condition is not first-order definable, $\mathsf{SQEMA}^{\mathsf{eagrec}}$ will fail on it.

Example 4.3 [Modified version of the Gödel-Löb axiom] Consider the following modified version of the Gödel-Löb axiom given by $\phi_2 := \Box(\Box p \wedge \Box\Box\bot \to p) \to \Box p$. Note that ϕ_2 is a recursive formula. From ϕ_2 we obtain one initial system namely:

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

Applying the $\wedge$-, $\Box$-, and left-shift-$\vee$-rules transforms this into (b) below, which in turn is transformed into (c) through the application of the recursive Ackermann rule:

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

The rest is routine.

After applying the recursive Ackermann rule, in parallel with the standard execution of $\mathsf{SQEMA}^{\mathsf{rec}}$ we can attempt to run the Eager Recursive SQEMA, by unfolding the fixed-point $\mu p.\Phi(p)$ with $\Phi(p) = \Diamond^{-1}\mathbf{i} \wedge \Box p \wedge \Box\Box\bot$. We find that $\Phi^n(\bot) = (\bigwedge_{i=0}^{n-1} \Box^i \Diamond^{-1}\mathbf{i}) \wedge \Box^2\bot \wedge \Box^n\bot \wedge \Box^{n+1}\bot$, for $n \geq 2$. But then for all $n \geq 2$ we have $\Phi^n(\bot) \equiv \Diamond^{-1}\mathbf{i} \wedge \Box\Diamond^{-1}\mathbf{i} \wedge \Box\Box\bot$. Thus, $\mathsf{SQEMA}^{\mathsf{eagrec}}$ detects that the fixed-point stabilizes at stage 2. The pure system obtained will thus have the form

$$(d)\ \|\mathbf{i} \Rightarrow \Diamond\neg(\Diamond^{-1}\mathbf{i} \wedge \Box\Diamond^{-1}\mathbf{i} \wedge \Box^2\bot \wedge \Box^2\bot \wedge \Box^3\bot).$$

In stage 3 we find that $\forall\overline{y}\exists x\mathrm{ST}(\neg\mathsf{pure}(\phi_2), x) = \exists x\mathrm{ST}(\mathbf{i} \wedge \Box(\Diamond^{-1}\mathbf{i} \wedge \Box\Diamond^{-1}\mathbf{i} \wedge \Box^2\bot \wedge \Box^2\bot \wedge \Box^3\bot), x)$, which simplifies to $\forall z(R^2 y_{\mathbf{i}} z \to R y_{\mathbf{i}} z) \wedge \neg\exists u R^3 y_{\mathbf{i}} u$. This corresponds to local transitivity of the current state and non-existence of successors more than 2 steps away.

Example 4.4 [Van Benthem's incomplete formula] Van Benthem's formula $\phi_{\mathsf{vB}} = \Box\Diamond\top \to \Box(\Box(\Box p \to p) \to p)$ is locally equivalent on frames to $\Box\Diamond\top \to \Box\bot$, and axiomatizes an incomplete modal logic ([31]). By simple equivalent transformations (replacing some implications with disjunctions) it is transformed into the recursive formula

$$\neg\Box\Diamond\top \vee \Box(\neg\Box(\Box p \to p) \vee p).$$

In phase 1 ϕ_{vB} is preprocessed to become $\Box\Diamond\top \wedge \Diamond(\Box(\Diamond\neg p \vee p) \wedge \neg p)$. In phase 2 a single initial system is constructed:

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

Through applications of the $\wedge$ and $\Diamond$-rules, it is transformed into (b) below, and then, by the application of the $\Box$ and left-shift-$\vee$-rules, into (c):

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

It should be clear that SQEMA would fail on this formula, since there is no possibility of separating the negative and positive occurrences of p in the last sequent. Indeed, it must fail, because SQEMA succeeds only on canonical formulae. However, the recursive Ackermann-rule can be applied to (c) to obtain:

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

The algorithm now performs phase 3 and returns a local FOμ frame equivalent for ϕ_{vB}.

By using some ad hoc reasoning we can derive from (d) the local condition $\Box\Diamond\top \to \Box\bot$ mentioned above. It suffices to show that $\mathbf{k} \to \neg\mu p.(\Diamond^{-1}\mathbf{k} \wedge \Box p)$ is a validity, and hence can be omitted from the system, thus producing from the resulting reduced system the pure formula $\mathbf{i} \to (\Box\Diamond\top \wedge \Diamond\mathbf{k})$ which is easily seen to yield the desired local condition. Equivalently, we must show that $\neg(\mathbf{k} \to \neg\mu p.(\Diamond^{-1}\mathbf{k} \wedge \Box p)) \equiv \mathbf{k} \wedge \mu p.(\Diamond^{-1}\mathbf{k} \wedge \Box p)$ is not satisfiable. For that, we will show that in any Kripke model the state s_k where $\mathbf{k}$ is evaluated cannot belong to the extension of $\mu p.(\Diamond^{-1}\mathbf{k} \wedge \Box p)$. Indeed, by transfinite induction we will prove that $s_k \notin \Phi^\alpha(\emptyset)$ for any ordinal α, where $\Phi(Z) = \Diamond^{-1}s_k \cap \Box Z$. The case of α a limit ordinal is straightforward, because then $\Phi^\alpha(\emptyset) = \bigcup_{\beta<\alpha}\Phi^\beta(\emptyset)$. For the case of $\alpha = \beta + 1$, let $X = \Phi^\beta(\emptyset)$. We will show, by contraposition, that if $s_k \in \Phi^\alpha(\emptyset) = \Phi(\Phi^\alpha(\emptyset)) = \Phi(X)$ then $s_k \in X$. Indeed, if $s_k \in \Diamond^{-1}s_k \cap \Box X$ then $s_k \in \Diamond^{-1}s_k$ hence s_k is reflexive. Therefore, $s_k \in \Box X$ implies $s_k \in X$.

Note, however, that $\mathsf{SQEMA}^{\mathsf{eagrec}}$ will fail to compute a first-order condition for ϕ_{vB}, since the fixed-point $\mu p.\Phi(p)$ does not stabilize in the finite. Thus, in order to succeed on ϕ_{vB}, the algorithm $\mathsf{SQEMA}^{\mathsf{rec}}$ must be extended with a mechanism that would detect the redundancy of the valid 3rd sequent in the system (d).

Example 4.5 [Grzegorczyk formula] Grzegorczyk's formula (also known as 'Dummett's axiom' in [31]) $\mathsf{Grz}(p) := \Box(\Box(p \to \Box p) \to p) \to p$ is a well known formula in modal logic, often related to the Gödel-Löb formula. We have seen that SQEMA succeeds on the Gödel-Löb formula, but, as we will show now, $\mathsf{SQEMA}^{\mathsf{rec}}$ fails on $\mathsf{Grz}(p)$.

Preprocessing yields $\Box(\Diamond(p \wedge \Diamond\neg p) \vee p) \wedge \neg p$. The initial system is thus $\| \ \mathbf{i} \Rightarrow \Box(\Diamond(p \wedge \Diamond\neg p) \vee p) \wedge \neg p$, which the application of the $\wedge$ and $\Box$-rules can transforms into

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

This system cannot be brought into the shape which makes even the monotonicity-based recursive Ackermann-rule (see section 6.2) applicable, since the negative and positive occurrences in the first sequent cannot be separated. The algorithm thus reports failure and terminates.

This failure does not imply that $\mathsf{Grz}(p)$ does not have a local condition in FOμ. In fact, one local condition for this formula is proposed in [31], p.48. It is a conjunction $\mathsf{GRZ}(x)'$ of the following three formulae:

$\mathsf{Ref(x)}$ Rxx — local reflexivity at x,

$\mathsf{Tr(x)}$ $\forall y \forall z(Rxy \wedge Ryz \to Rxz)$ — local transitivity at x, and

$\mathsf{Rchain'(x)}$ there is no R-chain $x = x_0 R x_1 \dots R x_n \dots$ such that $x_i \neq x_{i+1}$ for every $i = 0, 1, \dots$.

The following structure, however, provides a counterexample to the local frame equivalence of $\mathsf{Grz}(p)$ and this condition at x_0. Let
$W = \{x_0, x_1, x_2, x_3\}$ and let $R = \{(x_0, x_0), (x_0, x_1), (x_0, x_2), (x_0, x_3), (x_1, x_2), (x_2, x_3), (x_3, x_1)\}$.

It is easy to see that R satisfies the conditions $\mathsf{Ref}(x_0)$, $\mathsf{Tr}(x_0)$ but not $\mathsf{Rchain'}(x_0)$. Yet, $\mathsf{Grz}(p)$ is locally valid at x_0.

Instead of the condition $\mathsf{Rchain'}(x)$ consider the slightly modified condition:

$\mathsf{Rchain}(x)$ there is no R-chain $x = x_0 R x_1 \dots R x_n \dots$ such that $x_{2i} \neq x_{2j+1}$ for every $i, j = 0, 1, \dots$.

Now let $\mathsf{GRZ}(x)$ be the conjunction of $\mathsf{Ref}(x)$, $\mathsf{Tr}(x)$ and $\mathsf{Rchain}(x)$. Then it is not difficult to see that in any frame (W, R) and $x \in W$, $\mathsf{Grz}(p)$ is locally valid at x iff $\mathsf{GRZ}(x)$ holds.

The proof from right to left goes as follows. Suppose that $\mathsf{Grz}(p)$ is not true at x. Then one can construct by induction an infinite R-chain $x = x_0 R x_1 \dots R x_n \dots$ such that $x_{2i} \neq x_{2j+1}$ for every $i, j = 0, 1, \dots$, contradicting in this way condition $\mathsf{Rchain}(x)$.

For the direction from left to right we can reason by contraposition.

Case 1. Suppose that $\neg Rxx$. Define the valuation $V(p) = W - \{x\}$. This valuation falsifies $\mathsf{Grz}(p)$ at x.

Case 2. Suppose $\neg\mathsf{Tr}(x)$. Then there exist $x_1, x_2 \in W$ such that Rxx_1, Rx_1x_2, but $\neg Rxx_2$. Then $x \neq x_1 \neq x_2$. Define the valuation $V(p) = W - \{x, x_2\}$, and hence $x_1 \in V(p)$. This valuation falsifies $\mathsf{Grz}(p)$ at x.

Case 3. Suppose that $\neg\mathsf{Rchain}(x)$. Then there exists an R-chain $x = x_0 R x_1 \dots R x_n \dots$ such that $x_{2i} \neq x_{2j+1}$ for every $i, j = 0, 1, \dots$. Define the valuation $V(p) = W - \{x_{2i} \mid i = 0, 1, \dots\}$. Then we have that $x_{2j+1} \in V(p)$ for $j = 0, 1, \dots$. This valuation falsifies $\mathsf{Grz}(p)$ at x, which completes the proof.

Let us note that the global condition $(\forall x)\mathsf{GRZ'}(x)$ is a global equivalent of $\mathsf{Grz}(p)$, as it was noted in [3], page 137. This condition is expressible in FOμ by the following formula: $\forall x(xRx \wedge \forall y, z(xRy \wedge yRz \to yRz) \wedge \neg\nu(P, z).\Phi(P, z)[x])$, where $\Phi(P, z) = \exists w(zRw \wedge z \neq w \wedge P(w))$. To verify that this formula indeed defines condition $(\forall x)\mathsf{GRZ}'(x)$ it suffices to check that, for any frame $\mathfrak{F} = (W, R)$ and $w \in W$, if there is an infinite R-path $zw_1w_2\ldots$ starting from z with $z \neq w_1$ and $w_i \neq w_{i+1}$ for all i then $\mathfrak{F} \models \nu(P, z).\Phi(P, z)[z := w]$. Indeed, we have

$$\Phi^n(W) = \{z \in W \mid \exists w_1, \ldots, w_n(Rzw_1 \wedge \bigwedge_{1\leq i<n} Rw_iw_{i+1} \wedge z \neq w_1 \wedge \bigwedge_{1\leq i<n} w_i \neq w_{i+1})\},$$

and hence that

$$\Phi^\omega(W) = \bigcap_{i\in\omega} \Phi^i(W, z) = \{z \in W \mid \exists w_1, \ldots (Rzw_1 \wedge \bigwedge_{1\leq i} Rw_iw_{i+1} \wedge z \neq w_1 \wedge \bigwedge_{1\leq i} w_i \neq w_{i+1})\}.$$

Clearly $\Phi^{\omega+1}(W) = \Phi(\Phi^\omega(W)) = \Phi^\omega(W)$, i.e., the fixed-point stabilizes at $\omega + 1$. The desired result follows.

As far as local correspondence with FOμ is concerned, we have no proof yet that the local condition $\mathsf{GRZ}(x)$ of $\mathsf{Grz}(p)$ is expressible in FOμ, so we leave that as an open question.

5 Correctness, canonicity and scope of SQEMA$^{\mathsf{rec}}$

In this section we will establish three results. Firstly, we will prove that SQEMA$^{\mathsf{rec}}$ is correct in the sense that the returned FOμ-formula is always a local frame equivalent for the input. Secondly, we will show that all ML-formulae on which the eager version of SQEMA$^{\mathsf{rec}}$ succeeds are canonical and hence axiomatize complete modal logics. Thirdly, we will prove that SQEMA$^{\mathsf{rec}}$ successfully computes FOμ-equivalents for all recursive formulae. For the sake of the correctness and canonicity results we will need the following notions.

We begin by noting that, although SQEMA$^{\mathsf{rec}}$ takes input from ML, its execution invariably leads into the richer languages of HML and HMLμ. To cope with the possible shortage of singleton sets when interpreting the nominals of the latter language over descriptive frames, we define the following.

Definition 5.1 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 5.2 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 be a set of atoms (i.e., of propositional variables and/or nominals). We say that $\mathcal{M}$ and $\mathcal{M}'$ are AT-*related* if

(i) $V'(p) = V(p)$ or $V'(p) = W - V(p)$ for all propositional variables $p \in \mathsf{AT}$, and

(ii) $V'(\mathbf{j}) = V(\mathbf{j})$ for all nominals $\mathbf{j} \in \mathsf{AT}$.

The next definition is intended to capture two types of equivalence that can hold between the successive systems of sequents obtained during an execution of SQEMA$^{\mathsf{rec}}$. Hereafter, we denote the set of atoms occurring in a formula ϕ by $\mathsf{AT}(\phi)$.

Definition 5.3 Formulae $\phi, \psi \in \mathrm{HML}\mu$ are *transformation equivalent*, denoted $\phi \equiv_{\mathrm{tr}} \psi$, if for every model $\mathcal{M} = (\mathfrak{F}, V)$ such that $\mathcal{M} \Vdash \phi$ there exists an $(\mathsf{AT}(\phi) \cap \mathsf{AT}(\psi))$-related model $\mathcal{M}' = (\mathfrak{F}, V')$ such that $\mathcal{M}' \Vdash \psi$, and vice versa.

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

Definition 5.4 An execution of $\mathsf{SQEMA}^{\mathsf{rec}}$ is *sound on descriptive frames* if for every system of sequents Sys obtained during that execution and every system Sys' obtained from Sys by the application transformation rules, $\mathsf{Form}(\mathsf{Sys}) \equiv^{d}_{tr} \mathsf{Form}(\mathsf{Sys}')$.

Similarly, an execution of $\mathsf{SQEMA}^{\mathsf{rec}}$ is *sound on Kripke frames* if for every system of sequents Sys obtained and every system Sys' obtained from it by the application transformation rules, $\mathsf{Form}(\mathsf{Sys}) \equiv_{tr} \mathsf{Form}(\mathsf{Sys}')$.

5.1 Correctness

We will now establish the correctness of the algorithm in terms of local equivalence of the input modal formula to the returned $\mathrm{FO}\mu$-formula. The key to this result is the following lemma:

Lemma 5.5 *Every execution of $\mathsf{SQEMA}^{\mathsf{rec}}$ is sound on Kripke frames.*

Proof. It is sufficient to show that whenever a system Sys' results from a system Sys through the application of a $\mathsf{SQEMA}^{\mathsf{rec}}$ transformation rule, it is the case that $\mathsf{Form}(\mathsf{Sys}) \equiv_{tr} \mathsf{Form}(\mathsf{Sys}')$. The case of the recursive Ackermann-rule is justified by Lemma 4.1; the cases for all other transformation rules are the same as in [9]. □

The proof of the following theorem is, modulo reference to Lemma 5.5, essentially the same as that of the correctness part of Theorem 4.15 in [9], or as Theorem 4.3 in [10]:

Theorem 5.6 (Correctness of $\mathsf{SQEMA}^{\mathsf{rec}}$) *If $\mathsf{SQEMA}^{\mathsf{rec}}$ succeeds on a formula $\phi \in ML$ then the $FO\mu$-formula returned is a local frame correspondent of ϕ.*

5.2 Canonicity of $\mathsf{SQEMA}^{\mathsf{eagrec}}$-reducible ML-formulae

In [9] it was shown that all SQEMA-reducible formulae are canonical and hence axiomatize complete logics. In contrast, as we saw in example 4.2, $\mathsf{SQEMA}^{\mathsf{rec}}$ successfully terminates on some formulae, such as the Löb axiom, which are known to be non-canonical. Even worse, $\mathsf{SQEMA}^{\mathsf{rec}}$ succeeds on van Benthem's incomplete formula (example 4.4). It follows that no canonicity results are to be expected for the basic version of this algorithm. However, if we restrict our attention to the eager version, $\mathsf{SQEMA}^{\mathsf{eagrec}}$, we can indeed obtain a canonicity result.

Proposition 5.7 *If $\mathsf{SQEMA}^{\mathsf{rec}}$ succeeds on a ML-formula ϕ in such a way that the execution is sound both on descriptive and Kripke frames, then ϕ is d-persistent.*

Proof. Suppose $\mathsf{SQEMA}^{\mathsf{rec}}$ succeeds on $\phi \in \mathrm{ML}$ in such a way that the execution is sound both on descriptive and Kripke frames. 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 $\|\mathbf{i} \Rightarrow \neg\phi$, and Sys_m is the final, pure system and $\mathsf{pure}(\phi) = \mathsf{Form}(\mathsf{Sys}_m)$.

Let $\mathfrak{g} = (W, R, \mathbb{W})$ be a descriptive frame and $w \in W$. Then $(\mathfrak{g}, w) \Vdash \phi$ iff there is no augmented valuation V on $\mathfrak{g}$ such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}, V) \Vdash \neg\mathbf{i} \vee \neg\phi$. But, (since $\mathsf{Form}(\mathsf{Sys}_1) = \neg\mathbf{i} \vee \neg\phi$, and the execution 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 $(\mathfrak{g}, V) \Vdash \mathsf{pure}(\phi)$. Since nominals can range over all singletons, the latter is the case iff there is no valuation V on the underlying Kripke frame $\mathfrak{g}_\sharp$ of $\mathfrak{g}$, such that $V(\mathbf{i}) = \{w\}$ and $(\mathfrak{g}_\sharp, V) \Vdash \mathsf{pure}(\phi)$. 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 $(\mathfrak{g}_\sharp, V) \Vdash \neg\mathbf{i} \vee \neg\phi$. This is the case iff $(\mathfrak{g}_\sharp, w) \Vdash \phi$. $\square$

In [9] it is shown that the original algorithm SQEMA is sound on both descriptive and Kripke frames. 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. However, a restricted version does hold, the formulation of which requires the following definition:

Definition 5.8 A formula $\phi \in$ HML 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 5.9 (Ackermann's Lemma for Descriptive Frames [9]) *Suppose $A \in$ HML is a syntactically closed formula not containing p and $B(p) \in$ HML is a syntactically open formula which is negative in p. Then*

$$((A \to p) \wedge B(p)) \equiv^d_{\mathrm{tr}} B(A/p).$$

We are now ready to prove the main theorem of this subsection.

Theorem 5.10 *All $\mathsf{SQEMA}^{\mathsf{eagrec}}$-reducible ML-formulae are canonical.*

Proof. We begin by noting that no fixed-point operator is ever introduced during the execution of $\mathsf{SQEMA}^{\mathsf{eagrec}}$. Indeed, the only rule that could introduce such a operator is RAR. However, in $\mathsf{SQEMA}^{\mathsf{eagrec}}$ the expression $\mu p.A$ is replaced in the substitution by an equivalent one, obtained as some finite unfolding of the fixed-point. Hence the entire execution of $\mathsf{SQEMA}^{\mathsf{eagrec}}$ on a ML-input formula proceeds in the fragment HML.

Let us call sequents of the form $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$ introduced by the $\Diamond$-rule *diamond-link sequents*. An easy induction on the application of transformation rules shows that, apart from diamond-link sequents, all sequents $\alpha \Rightarrow \beta$ obtained during the execution of $\mathsf{SQEMA}^{\mathsf{eagrec}}$ are always such that α is syntactically closed and β is syntactically open. Also note that, because they are pure, diamond-link sequents can never be involved in the substitutions performed during any application of RAR.

Combining these observations with Lemma 5.9 we find that any application of RAR by $\mathsf{SQEMA}^{\mathsf{eagrec}}$ preserves transformation equivalence over descriptive frames. That it also preserves transformation equivalence on Kripke frames follows by Lemma 4.1. From this and the easily verifiable fact that all other rules also preserve these equivalences, it follows that all executions of $\mathsf{SQEMA}^{\mathsf{eagrec}}$ are sound on both descriptive and Kripke frames. The theorem now follows by Proposition 5.7. $\square$

Example 5.11 We saw in example 4.3 that $\mathsf{SQEMA}^{\mathsf{eagrec}}$ successfully computes a local frame equivalent for the modified Gödel-Löb formula $\Box(\Box p \wedge \Box\Box\bot \to p) \to \Box p$. By Theorem 5.10 we can now also conclude that this formula is canonical.

5.3 Completeness of $\mathsf{SQEMA}^{\mathsf{rec}}$ for all recursive formulae

The following theorem illustrates the scope of the $\mathsf{SQEMA}^{\mathsf{rec}}$-algorithm.

Theorem 5.12 *$\mathsf{SQEMA}^{\mathsf{rec}}$ succeeds on all recursive formulae.*

Proof. First, note that the negation of any recursive formula is built, up to semantic equivalence, from conjunctions, disjunctions, and diamonds applied to negative formulae and box-formulae. Preprocessing transforms such a formula into a disjunction of formulae built from negative formulae and box-formulae using conjunctions and diamonds. By successive application of the $\wedge$-rule and $\Diamond$-rule, $\mathsf{SQEMA}^{\mathsf{rec}}$ will strip off all diamonds and conjunctions and will produce a system of equations of 3 types:

(i) $\mathbf{j} \Rightarrow \Diamond\mathbf{k}$,

(ii) $\mathbf{j} \Rightarrow \beta$, where β is a negative formula,

(iii) $\mathbf{j} \Rightarrow \alpha$, where α is a box-formula (rewritten in negation normal form).

Now, it suffices to process every formula of the third type by applying successively, on the construction of the respective box-formula, the $\Box$-rule and the left-shift $\vee$-rule (after replacing all implications by disjunctions), until the respective sequent reaches the shape $POS \Rightarrow p$, where p is the head of the processed box-formula and POS is a positive formula. Indeed, all formulae that go to the left-hand side in that process are antecedents of implications used in the construction of the box-formula, to which inverse diamonds then get applied.

After every sequent of type 3 has been processed, the system becomes:

$$\left\| \begin{array}{l} A_1(p_1, \ldots, p_n) \Rightarrow p_1 \\ \ldots \\ A_n(p_1, \ldots, p_n) \Rightarrow p_n \\ B_1(p_1, \ldots, p_n) \\ \ldots \\ B_m(p_1, \ldots, p_n) \\ C_1 \\ \ldots \\ C_k \end{array} \right.$$

where

(i) $p_1, \ldots, p_n$ are variables, that can be assumed different,

(ii) $A_1, \ldots, A_n$ are positive formulae over $p_1, \ldots, p_n$,

(iii) $B_1, \ldots, B_m$ are negative formulae over $p_1, \ldots, p_n$, and

(iv) $C_1, \ldots, C_k$ are pure hybrid formulae.

Now, all variables can be eliminated from such system by applying RAR, thus completing the execution of $\mathsf{SQEMA}^{\mathsf{rec}}$. □

6 Further extensions of the algorithm

$\mathsf{SQEMA}^{\mathsf{rec}}$ is amenable to various extensions and variations, of which we will briefly present three.

6.1 Polyadic extensions of $\mathsf{SQEMA}^{\mathsf{rec}}$

The polyadic version of the recursive formulae was defined in [21], where it was shown that all polyadic recursive formulae (there called ‘regular formulae’) are locally FOμ-definable. A simple example of a polyadic recursive formula which is not inductive is $[\iota_2](\neg[\mathbf{2}](p,p), \langle\mathbf{2}\rangle(p,p))$. It defines the following non-elementary frame condition on frames with one ternary relation R: *“For every x the binary relation R_x on the remaining two variables y and z has an unoriented cycle of odd length.”*

The polyadic SQEMA was developed in [10]. The recursive extension of the Ackermann rule for the polyadic SQEMA is essentially analogous to the monadic case, and most of the results presented here generalize without any essential difficulty beyond the technical overhead of considering polyadic languages. It must be noted, however, that the proof of the canonicity result for polyadic SQEMA is significantly more involved than in the monadic case. Accordingly it is to be expected that a proof that all formulae reducible by the polyadic version of $\mathsf{SQEMA}^{\mathsf{eagrec}}$ are canonical would involve similar complications.

6.2 Semantic extensions

An obvious generalization of the (ordinary) modal version of Ackermann's lemma (see e.g. [9]) replaces the requirements of positivity and negativity on the formulae involved with their semantic correlates, namely upward and downward monotonicity, respectively. Different 'semantic' versions of SQEMA which arise from the replacement of the ordinary Ackermann-rule with a monotonicity based version, justified by this generalization of the Ackermann lemma, have been developed in [7].

If one is willing to extend the syntax of HMLμ to allow for fixed-point operators over upward monotone formulae, one can obtain a similar generalization of the recursive Ackermann rule RAR to a monotonicity based version MRAR. Let us call the extension of $\mathsf{SQEMA}^{\mathsf{rec}}$ with this rule $\mathsf{SQEMA}^{\mathsf{monrec}}$. Of course, for a rule like MRAR to be of any practical use one will have to be able to effectively test HMLμ-formulae for monotonicity. By the following lemma, the proof of which is almost immediate, one may effectively reduce testing for monotonicity to testing satisfiability.

Lemma 6.1 *An HMLμ-formula $\phi(p)$ is downwards monotone in p iff $\Vdash \phi(p) \rightarrow \phi(p \wedge q)$ where q is any variable not occurring in $\phi(p)$.*

Using the decidability of the satisfiability for HMLμ from [27], the applicability of MRAR may be effectively tested, and moreover our extension of HMLμ allowing fixed-point operators over monotone formulae is justified in the sense that the syntax would remain decidable.

As far as the canonicity result for $\mathsf{SQEMA}^{\mathsf{eagrec}}$ is concerned we recall that the proof of this result depended on the syntactic shape (viz. syntactic openness and closedness) of formulae involved in the application of RAR. After one application of MRAR this shape of the remaining sequents may be spoiled, and hence subsequent applications of MRAR may not fall under Lemma 5.9. It is not clear at present if and how this difficulty can be circumvented and hence we leave it as an open question whether all formulae on which the semantic version of $\mathsf{SQEMA}^{\mathsf{eagrec}}$ succeeds are canonical. In [7] some restrictions and modifications of the semantic algorithm which would guarantee canonicity were proposed. It should be possible to obtain a similar effect by making analogous modifications to a semantic version of $\mathsf{SQEMA}^{\mathsf{eagrec}}$.

6.3 Recursive extensions of SQEMA with substitutions

In the recent work [11], SQEMA was extended with a mechanism for computing and applying suitable substitutions that transform an input formula into an inductive one, and thus enable the algorithm to successfully complete its FO-equivalent. In particular, the so extended algorithm succeeds on all 'complex formulae', introduced in [28].

Likewise, there are many non-recursive modal formulae that can be transformed by means of substitutions to ones amenable to applying the recursive Ackermann rule. That idea can be seamlessly incorporated into the recursive extensions of SQEMA, thus expanding their scope further and, in particular, enabling them to deal with 'complex recursive formulae' [11]. The following is an interesting example of a complex recursive formula which is not a recursive one (see [30]):

$$\left(\begin{array}{c}\langle U\rangle(\neg p \wedge \neg q) \wedge \langle U\rangle(\neg p \wedge q) \wedge \langle U\rangle(p \wedge \neg q) \wedge \langle U\rangle(p \wedge q) \\ \wedge[U](\Diamond(p \vee q) \rightarrow (p \vee q)) \wedge [U](\Diamond p \rightarrow (p \vee \neg q)) \wedge [U](\Diamond(p \wedge q) \rightarrow (\neg p \vee q))\end{array}\right) \rightarrow \bot$$

The above formula has a local equivalent in FOμ, which in symmetric Kripke frames (W, R) (considered as graphs) with $[U]$ interpreted as the universal modality and $\langle U\rangle$ as the corresponding diamond modality, has the following global equivalent (C3) which says that the graph (W, R) has at most three connected components:

$$(\forall x_0, x_1, x_2, x_3)(\exists k \geq 0)(\exists i, j)(0 \leq i < j \leq 3)(x_i R^k x_j) \tag{C3}$$

For instance, the connectedness of a symmetric frame (W, R) can be expressed by the following condition (expressible also in FOμ)

$$(\forall x, y)(\exists k)(x R^k y) \tag{C1}$$

(C1) is modally definable (in symmetric frames) by the following recursive formula

$$\langle U\rangle\neg p \wedge \langle U\rangle p \wedge [U](\Diamond p \Rightarrow p) \Rightarrow \bot.$$

7 Conclusion

We have developed extensions of the algorithm SQEMA employing a recursive version of the Ackermann-rule, that can be used for computing FOμ-equivalents of modal formulae, thus proposing an algorithmic approach to the correspondence theory between modal logic and FOμ. We have demonstrated the applicability of the proposed method on a number of well known and new modal formulae defining non-elementary frame conditions. A number of problems and directions for further investigation remain open. Perhaps the most intriguing of them is if the notion of canonicity can be appropriately relaxed so that a suitable version of SQEMA$^{\text{rec}}$ can not only compute FOμ-correspondents of input formulae, but also establish (non-canonical) completeness of the modal logics axiomatized by those input formulae on which it succeeds.

Acknowledgements

We gratefully acknowledge the financial support of the National Research Foundation of South Africa through grant number NRF UID 70554. Dimiter Vakarelov was supported by contract 063/2009 with Sofia University.

Appendix: The basic algorithm SQEMA

Here we present very briefly the basic algorithm SQEMA for the reader's convenience; for more detail, see [9].

First, some terminology — an expression of the form $\phi \Rightarrow \psi$ with $\phi, \psi \in \mathcal{L}_r^n$ is called a *SQEMA-sequent* [5], 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 $\mathsf{Form}(\phi \Rightarrow \psi) := \neg\phi \vee \psi$ and, for a system Sys, we let $\mathsf{Form}(\mathsf{Sys})$ be the conjunction of all $\mathsf{Form}(\phi_i \Rightarrow \psi_i)$ for all sequents $\phi_i \Rightarrow \psi_i \in \mathsf{Sys}$.

Given a formula $\phi \in \mathcal{L}$ as input, SQEMA 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 $\bigvee \phi_i'$ a system Sys_i is formed consisting of the single sequent $\mathbf{i} \Rightarrow \phi_i'$, where $\mathbf{i}$ is a reserved nominal used to denote the state of evaluation in a model, and not allowed to occur in the input formula ϕ. 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 Ackermann-rule, systems with new systems. Note that each actual elimination of a variable is achieved through an application of the 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 Ackermann-rule can be determined with the help of 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 $\mathsf{pure}(\phi)$ 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 $\forall\overline{y}\exists x_0 \mathrm{ST}(\neg\mathsf{pure}(\phi), x_0)$ 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 SQEMA succeeds will be called *SQEMA-reducible*, or simply *reducible*.

Finally, to note that the algorithm can be strengthened further by adding more transformation rules facilitating some propositional reasoning, see [9].

[5] In [9] sequents are called ‘equations’ because of the analogy with solving systems of linear equations.

Table 1
SQEMA Transformation Rules

Rules for connectives

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

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

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

$$\frac{\mathbf{j} \Rightarrow \Diamond A}{\mathbf{j} \Rightarrow \Diamond \mathbf{k},\ \mathbf{k} \Rightarrow A}\ (\Diamond\text{-rule}^*)$$

∗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.

Ackermann-rule

The system
$$\left\| \begin{array}{l} A \Rightarrow p \\ B_1(p) \\ \dots \\ B_m(p) \\ C_1 \\ \dots \\ C_k \end{array} \right.$$
is replaced by
$$\left\| \begin{array}{l} B_1(A/p) \\ \vdots \\ B_m(A/p) \\ \\ C_1 \\ \dots \\ C_k \end{array} \right.$$

where:

(i) p does not occur in $A, C_1, \dots, C_k$;

(ii) $\mathsf{Form}(B_1) \wedge \cdots \wedge \mathsf{Form}(B_m)$ is negative in p.

(We have added the $\vee$-rule to the system in order to simplify the Ackermann rule from [9] by enabling all sequents of the type $A \Rightarrow p$ to be put together.)

References

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

[2] Arnold, A. and D. Niwinski, "Rudiments of μ-Calculus," North-Holland, 2001.

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

[4] Blackburn, P., J. van Benthem and F. Wolter, editors, "Handbook of Modal Logic," Elsevier, 2006.

[5] Blass, A., Y. Gurevich and D. Kozen, *A zero-one law for a logic with a fixed-point operator*, Information and Control **67** (1995), pp. 70–90.

[6] Bradfield, J. and C. Stirling, *Modal μ-calculi*, in: Blackburn et al. [4] pp. 729–756.

[7] Conradie, W. and V. Goranko, *Algorithmic correspondence and completeness in modal logic IV. Semantic extensions of* SQEMA, Journal of Applied Non-Classical Logics **18(2-3)** (2008), pp. 175–211.

[8] Conradie, W., V. Goranko and D. Vakarelov, *Elementary canonical formulae: a survey on syntactic, algorithmic, and model-theoretic aspects*, , **5** (2005), pp. 17–51.

[9] Conradie, W., V. Goranko and D. Vakarelov, *Algorithmic correspondence and completeness in modal logic I: The core algorithm* SQEMA, Logical Methods in Computer Science **2(1:5)** (2006).

[10] Conradie, W., V. Goranko and D. Vakarelov, *Algorithmic correspondence and completeness in modal logic II. Polyadic and hybrid extensions of the algorithm* SQEMA, Journal of Logic and Computation **16** (2006), pp. 579–612.

[11] Conradie, W., V. Goranko and D. Vakarelov, *Algorithmic correspondence and completeness in modal logic III.* SQEMA *with substitutions*, Submitted (2008).

[12] Conradie, W. and G. van Drimmelen, *Deciding boundedness for the full μ-calculus* (2008), manuscript.

[13] Doherty, P., W. Łukaszewicz and A. Szalas, *Computing circumscription revisited: A reduction algorithm*, Journal of Automated Reasoning **18** (1997), pp. 297–336.

[14] Ebbinghaus, H.-D. and J. Flum, "Finite Model Theory," Perspectives in Mathematical Logic, Springer, Berlin, 1995.

[15] Flum, J., *On the (infinite) model theory of fixed-point logics*, Preprint, Inst. für Math. Logik und Grundl. der Math., Albert-Ludwigs-Universität Freiburg (1995).
URL http://logimac.mathematik.uni-freiburg.de/preprints/flu95.dvi

[16] Gabbay, D. M., R. Schmidt and A. Szalas, "Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications," Studies in Logic: Mathematical Logic and Foundations, College Publications, 2008.

[17] Goranko, V. and B. Kapron, *The modal logic of the countable random frame*, Arch. Math. Logic **42** (2003), pp. 221–243.

[18] Goranko, V. and M. Otto, *Model theory of modal logic*, in: Blackburn et al. [4] pp. 250–329.

[19] Goranko, V. and D. Vakarelov, *Sahlqvist formulae in hybrid polyadic modal languages*, Journal of Logic and Computation **11(5)** (2001), pp. 737–254.

[20] Goranko, V. and D. Vakarelov, *Sahlqvist formulas unleashed in polyadic modal languages*, , **3** (2002), pp. 221–240.

[21] Goranko, V. and D. Vakarelov, *Elementary canonical formulae: Extending Sahlqvist theorem*, Annals of Pure and Applied Logic **141(1-2)** (2006), pp. 180–217.

[22] Le Bars, J., *Zero-one law fails for frame satisfiability in propositional modal logic*, in: *Proceedings of LICS'2002* (2002), pp. 225–234.

[23] Libkin, L., "Elements of Finite Model Theory," Springer, Berlin, 2004.

[24] Nonnengart, A. and A. Szałas, *A fixpoint approach to second-order quantifier elimination with applications to correspondence theory*, in: E. Orłowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, Studies in Fuzziness and Soft Computing **24** (1998), pp. 307–328.

[25] Otto, M., *The boundedness problem for monadic universal first-order logic*, in: *Proceedings of 21st IEEE Symposium on Logic in Computer Science LICS'06*, 2006, pp. 37–46.

[26] Sahlqvist, H., *Correspondence and completeness in the first and second-order semantics for modal logic*, in: S. Kanger, editor, *Proceedings of the 3rd Scandinavian Logic Symposium, Uppsala 1973* (1975), pp. 110–143.

[27] Sattler, U. and M. Y. Vardi, *The hybrid μ-calculus*, in: *IJCAR*, 2001, pp. 76–91.

[28] Vakarelov, D., *Modal definability in languages with a finite number of propositional variables, and a new extension of the Sahlqvist class*, in: F. W. P. Balbiani, N.-Y. Suzuki and M. Zakharyaschev, editors, *Advances in Modal Logic, vol. 4* (2003), pp. 495–518.

[29] Vakarelov, D., *Solving recursive equations in complete modal algebras with applications to modal definability*, in: *Pioneers of Bulgarian Mathematics, International Conference dedicated to Nicola Obrechkoff and Lubomir Tschakaloff, Sofia, July 8–10*, 2006.

[30] Vakarelov, D., *A recursive generalizations of ackermann lemma with applications to μ-definability*, in: G. Kaouri and S. Zahos, editors, *Proc. of the 6th Panhellenic Logic Symposium PLS-2007*, 2007, pp. 133–137.

[31] van Benthem, J. F. A. K., "Modal Logic and Classical Logic," Bibliopolis, 1983.

[32] van Benthem, J. F. A. K., *Correspondence theory*, in: D. Gabbay and F. Guenthner, editors, *Handbook of Phil. Logic, vol. II*, Reidel, Dordrecht, 1984 pp. 167–247.

[33] van Benthem, J. F. A. K., *Minimal predicates, fixed-points, and definability*, Journal of Symbolic Logic **70:3** (2005), pp. 696–712.

[34] van Benthem, J. F. A. K., *Modal frame correspondence and fixed-points*, Studia Logica **83** (2006), pp. 133–155.