A Fixpoint Approach to Second-Order Quantifier Elimination with Applications to Correspondence Theory

Andreas Nonnengart[*]
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken, Germany
e-mail: nonnenga@mpi-sb.mpg.de

Andrzej Szałas[†]
Institute of Informatics
University of Warsaw
ul. Banacha 2, 02-097 Warsaw, Poland
e-mail: szalas@mimuw.edu.pl

Abstract

This paper is about automated techniques for second-order quantifier elimination and (modal logic) correspondence theory. Given a modal schema and a semantics based method of translating modal formulae into fist-order ones, we try to derive automatically a fixpoint formula characterizing precisely the class of frames validating this schema. We propose to consider fixpoint calculus as the underlying formalism rather than the classical first-order logic in order to be able to express more subtle correspondences. The technique we consider can, in many cases, be easily applied without any computer support.

Although we mainly concentrate on Kripke semantics, our fixpoint approach is much more general, as it is based on the elimination of second-order quantifiers from formulae. Thus it can be applied in second-order theorem proving as well. We show some application examples for the method which may serve as new, automated proofs of the respective correspondences.

Keywords: correspondence theory, fixpoint calculus, modal logics, quantifier elimination, second-order logic, semantics based translation

1 Introduction

The development of possible world semantics in the late fifties and the early sixties opened the area of the so-called *Correspondence Theory*, i.e. the discovery of strict connections between Hilbert style axioms and properties of the accessibility relation between worlds. For instance it was shown that the modal logic *T-axiom* $\Box\Phi \supset \Phi$ is valid in a modal frame with accessibility relation $\Re$ if and only if $\Re$ is reflexive. Many such correspondences have been discovered since then and it became of major interest in this area to develop methods which allow to find such correspondences and which provide a formal basis for doing so. (As a main sourcebook to this see [22]). Until recently, all of these methods were based on more or less abstract model

[*]Supported in part by the German Ministry for Research and Technology (Bundesministerium für Forschung und Technologie) under grant ITS 9102.

[†]Supported in part by the KBN grant 3 P406 019 06

theoretic considerations and it can hardly be imagined that these can be automated. In the last couple of years, however, major steps have been taken towards an automated approach of finding modal logic correspondences. For example we can find algorithms for an automatic synthesis of correspondence axioms expressed in classical first-order logic in [5, 15, 18, 19] where the algorithm of [18, 19] is further developed in [4]. All these approaches are incomplete. On the other hand, we know that the problem we deal with is totally undecidable (not even arithmetical - cf. [22]). Some of these algorithms (the ones presented in [4, 15, 18, 19]) always stop, while that of [5], called SCAN, sometimes loops. However, when SCAN loops, it is often possible to observe certain patterns which, put together, provide us with a correspondence axiom described in infinitary logic. This method is similar to Ackermann's general method of second-order quantifier elimination given in [1], which works fine in general but, unfortunately, is difficult to use and hard to apply for it deals with infinitary logic. Fragments of $L_{\omega_1\omega}$ are also applied in this context in [14].

With the approach proposed in this paper we try to avoid these problems if possible, i.e. we want to get answers to correspondence problems for which SCAN loops and the general Ackermann technique results in an infinite formula. Evidently, infinite formulae are problematic for any automated approach and therefore these have to be finitely encoded somehow. We do so by allowing fixpoint operators within formulae. We thus deal with a formalism which still has a nice proof theory (cf. [8, 16, 17]) and polynomial time model checking over finite domains (cf. [7]). The main goals of the paper are then the following:

- to present an automated method of reducing second-order quantifiers into the fixpoint formulas
- to show that the method we propose is important in correspondence theory and extends known approaches
- to propose fixpoint calculus as a logic suitable to express semantical properties induced by Hilbert-like modal axioms.

The paper is organized as follows: Section 2 contains the basic principles behind modal logics, their semantics based (relational) translation into first-order predicate logic, and earlier second-order quantifier elimination techniques. Moreover, we briefly describe the major features of the fixpoint calculus and that just as far as they are necessary in order to be able to understand our main theorem given in Section 3. In Section 4 we apply this technique to a number of examples, including interesting modal logic axiom schemata which could not be solved with the automated approaches known so far. A final comparisons of results and summary then conclude the paper.

2 Preliminaries

2.1 Notation

In what follows Φ and Ψ are meta-variables representing formulae, $\wedge$, $\vee$, $\supset$ and $\equiv$ stand for conjunction, disjunction, implication and equivalence respectively. The negation is denoted by $\overline{\Phi}$; $\bar{x}$ and $\bar{y}$ stand for tuples of variables. By $\top$ and $\bot$ we shall denote the truth values *true* and *false*, respectively. Predicates and predicate variables are denoted by P, Q, ... The modal operators are denoted by $\Box_i$ and $\Diamond_i$, where $i \in I$ for some index set I. By $A \leftarrow B$ we denote the substitution of expression A by expression B.

2.2 The SCAN Algorithm

The SCAN algorithm (short for "Synthesizing Correspondence Axioms for Normal modal logics") is a method to eliminate second-order quantifiers in order to find first-order equivalents if these exist. The main idea behind this algorithm is as follows: suppose there is the (second-order) formula $\Phi = \exists P\ \Psi$ where Ψ is a first-order subformula and we want to find a first-order equivalent to Φ. Then transform Ψ into clause normal form (i.e. negation normal form and standard skolemization) and perform all possible *constraint resolution* and *constraint factorization steps* that are possible between P-literals. By a *constraint resolution step* it is meant to perform a resolution step even if the corresponding arguments are not unifiable. This means that residual inequalities have to be added to the resolvents sometimes. So, for example, a constraint resolution step between the two unit clauses $P(a)$ and $\overline{P(b)}$ results in the (constraint) resolvent $a \neq b$ (and similarly for constraint factorization steps). Very often this process terminates since after all possible constraint inference steps are performed with some literal $P(\alpha)$ in a clause C this very clause can be eliminated by purity deletion. Thus, after termination, the algorithm ends up with a set of clauses not containing any P-literal. A further necessary step in this approach is to deskolemize the clause set, i.e. all function symbols which were actually introduced by the skolemization during the clause form transformation have to be "back-translated" into existential quantifiers and if this succeeds we end up with a first-order formula which is equivalent to the second-order formula given as input.

Example 2.1
Consider the following simple formula

$$\forall P\ [(\forall x\ P(x) \supset Q(x)) \ \supset\ \forall x\ \overline{P(x)}] \tag{1}$$

which states that every subset of the set Q is empty. Evidently this is equivalent to Q being empty and SCAN indeed has no problems with it. (Note that this formula has a universal second-order quantifier; it therefore has to be negated beforehand in order to be able to apply the SCAN algorithm. This evidently does not affect the whole procedure for the final result can be negated once again.) Hence we first negate the input formula, getting $\exists P\ [(\forall x\ P(x) \supset Q(x)) \wedge \exists x\ P(x)]$ and try to find a first-order equivalent for this one. Transformation into clause normal form results in

$$\begin{array}{l} \overline{P(x)}, Q(x) \\ P(a) \end{array}$$

and the task is now to produce all possible constraint resolvents from this clause set. As a matter of fact, this is trivial because only one resolution step is at all possible. This step results in the resolvent $Q(a)$ and since all possible steps between P-literals have been performed now the clauses containing P-literals can be eliminated leaving the unit clause $Q(a)$ as the only remaining clause. Recall that the constant a had been introduced by skolemization. It therefore has to be back-translated into an existential quantifier and after doing so we end up with $\exists x\ Q(x)$.
We thus have that $\exists P\ [(\forall x\ P(x) \supset Q(x)) \wedge \exists x\ P(x)]$ is equivalent to $\exists x\ Q(x)$ and therefore

$$\forall P\ [(\forall x\ P(x) \supset Q(x)) \ \supset\ \forall x\ \overline{P(x)}] \quad \equiv \quad \forall x\ \overline{Q(x)}$$

SCAN has no problems with this example. Nevertheless, already a tiny change lets SCAN run into an infinite loop as can be seen as follows: consider the formula

$$\forall P\ [(\forall x\ P(x) \supset \exists y\ P(y) \wedge Q(x)) \supset \forall x\ \overline{P(x)}] \tag{2}$$

which tells us just the same as the formula (1); it only contains some superfluous ballast given by the $\exists y\, P(y)$. Now, after transforming the negation of (2) into clause normal form we get:

$$\begin{array}{l}\overline{P(x)}, P(f(x))\\ \overline{P(x)}, Q(x)\\ P(a)\end{array}$$

SCAN now attempts to perform all possible constraint resolution steps between P-literals. However, the first and third of these three clauses allow to derive infinitely many unit P-clauses. SCAN tries to produce all of them and therefore never terminates.

However, if we take a look at the (infinitely many) resolvents produced by SCAN we can find a fairly nice pattern of resolvents, namely the unit clauses $Q(f^n(a))$ and $P(f^n(a))$ with $n \geq 0$. Thus, if SCAN were able to find such clause schemata it would be sufficient to find a first-order equivalent to $\exists a \exists f\, Q(f^n(a))$ and this can be found in $\exists a\, Q(a)$ just as expected. ∎

Unfortunately, SCAN is not able to derive such clause schemata and therefore the final steps of the above example cannot be automated yet. We therefore concentrate on another approach on second-order variable elimination, namely the one presented in [19, 4].

2.3 The Ackermann Lemma

Note that the method we propose extends known approaches, in particular the one based on the lemma of Ackermann (cf. [18, 19]). This lemma is a part of the Ackermann's general method given in [1]. It is interesting, as the result is always first-order. We therefore start with a formulation of this lemma.

We say that a formula Φ is *positive* w.r.t. a predicate P iff P appears under no negation sign in Φ (in the negation normal form). Dually, we say that Φ is *negative* w.r.t. P iff all occurrences of P have the form $\overline{P}$ and $\overline{P}$ appears under no negation sign in Φ.
The following lemma was proved by Ackermann in [1].

Lemma 2.2
Let P be a predicate variable and $\Phi(\bar{x})$, $\Psi(P)$ be formulae without second-order quantification. Let $\Psi(P)$ be positive w.r.t. P and let Φ contain no occurrences of P at all. Then the following equivalences hold:

$$\exists P \forall \bar{x}[P(\bar{x}) \vee \Phi(\bar{x}, \bar{z})] \wedge \Psi(P \leftarrow \overline{P}) \;\equiv\; \Psi(P \leftarrow \Phi(\bar{x}, \bar{z})) \tag{3}$$

$$\exists P \forall \bar{x}[\overline{P}(\bar{x}) \vee \Phi(\bar{x}, \bar{z})] \wedge \Psi(P) \;\equiv\; \Psi(P \leftarrow \Phi(\bar{x}, \bar{z})) \tag{4}$$

where in the righthand formulae the arguments $\bar{x}$ of Φ are to be substituted by the respective actual arguments of P (renaming the bound variables whenever necessary). ∎

The syntactic form required in Lemma 2.2 is usually not given initially and we are therefore forced to apply some well-known equivalence preserving transformations of classical first-order logic in order to obtain this form. The following one was found particularly useful in [1]:

$$P(\bar{x}) \;\equiv\; \forall \bar{y}\, (P(\bar{y}) \vee \bar{x} \neq \bar{y}). \tag{5}$$

This technique was substantially strengthened in [4] by applying the following equivalence:

$$P(\overline{x_1}) \vee P(\overline{x_2}) \vee \cdots \vee P(\overline{x_n}) \;\equiv\; \exists \bar{z}\, [(\bar{z} = \overline{x_1} \vee \bar{z} = \overline{x_2} \vee \cdots \vee \bar{z} = \overline{x_n}) \wedge P(\bar{z})] \tag{6}$$

Other quite well-known transformation techniques turn out to be useful as well as there are e.g. the transformation into conjunctive normal form and the following second-order skolemization that preserves equivalence of formulae and allows to eliminate existential quantifiers (similar formulation can be found e.g. in [21]):

$$\forall \bar{x} \exists y \Phi(\bar{x}, y, \ldots) \;\equiv\; \exists f \forall \bar{x} \Phi(\bar{x}, y \leftarrow f(\bar{x}), \ldots). \tag{7}$$

One of the main restrictions in Lemma 2.2 is that it does not allow clauses which contain both, positive and negative occurrences of the predicate symbol to be eliminated. In order to overcome this difficulty we prove Theorem 3.1, which shows how to deal with such a situation. Moreover, Theorem 3.1 subsumes Lemma 2.2. We thus substantially extend the approach based on the Ackermann lemma.

2.4 Fixpoint Calculus

Let $\mathcal{L}_\mathcal{I}$ be the classical first-order logic. In order to define the *fixpoint calculus* $\mathcal{L}_\mathcal{F}$ we extend $\mathcal{L}_\mathcal{I}$ by allowing the least fixpoint operator $\mu P.\Phi(P)$, where Φ is positive w.r.t. P and abbreviate a formula of the form $\overline{\mu \overline{P}.\overline{\Phi}(\overline{P})}$ by $\nu P.\Phi(P)$. It is sometimes convenient to indicate the individual variables that are bound by the fixpoint operators μ and ν. We write $\mu.P(\bar{x})$ and $\nu.P(\bar{x})$ to indicate that the tuple $\bar{x}$ of variables is bound by a fixpoint operator. Example 2.3 explains the rôle of fixpoint operators and the variables bound by μ and ν.

Let us now recall some useful well-known facts. The formulation we give is adapted to the particular problems we deal with. In particular, the partial order we consider is the following:

- the carrier is the set of formulae of $\mathcal{L}_\mathcal{F}/_\equiv$, where we do not distinguish between logically equivalent formulae (formally the carrier is the quotient set $\mathcal{L}_\mathcal{F}/_\equiv$; in order to simplify the considerations the equivalence classes and formulae are identified)
- the formulae are ordered by implication, i.e. Φ is less or equal to Ψ iff $\Phi \supset \Psi$ is a tautology.

Note that fixpoint operators are also formulae. Thus the partial order we consider is complete. Every formula $\Phi(P)$ which is positive w.r.t. P is monotone and therefore we have by the Knaster & Tarski fixpoint theorem that the fixpoints we consider are well defined. Moreover, the fixpoints have the following nice characterization:

$$\mu P.\Phi(P) \;\equiv\; \bigvee_{\beta \in \alpha} \Phi^\beta(\bot) \tag{8}$$

for an ordinal α (the least such ordinal is called the *closure ordinal for* $\Phi(P)$).

Note that $\mu P(\bar{x}).\Phi(P)$ is the least (w.r.t. the partial order defined above) formula $\Psi(\bar{x})$ such that

$$\Psi(\bar{x}) \;\equiv\; \Phi(P \leftarrow \Psi(\bar{x})).$$

We illustrate the introduced notions by the following example:

Example 2.3
Consider $\mu P(x).\Phi(P)$, where $\Phi(P(x)) \equiv [x = y \vee \exists z\ (R(x,z) \wedge P(z))]$. Then the formula $\Phi(P(x))$ is positive w.r.t. P. Moreover, it is possible to show that the respective closure ordinal is ω (see e.g. the proof of Theorem 3.1). Thus, according to (8),

$$\mu P(x).\ [x = y \vee \exists z\ (R(x,z) \wedge P(z))] \quad \equiv \quad [\Phi(P)]^{\omega}(\bot) \quad \equiv \quad \bigvee_{i \in \omega} [\Phi(P)]^{i}(\bot)$$

where

$$\begin{array}{lcl}
\Phi^0(\bot) & \equiv & \bot \\
\Phi^1(\bot) & \equiv & [x = y \vee \exists z\ (R(x,z) \wedge P(z))]^1(\bot) \\
 & \equiv & x = y \vee \exists z\ (R(x,z) \wedge \Phi^0(\bot)) \\
 & \equiv & x = y \\
\Phi^2(\bot) & \equiv & \Phi(\Phi^1(\bot)) \\
 & \equiv & \Phi(x = y) \\
 & \equiv & x = y \vee \exists z\ (R(x,z) \wedge z = y) \\
 & \equiv & x = y \vee R(x,y) \\
\Phi^3(\bot) & \equiv & \Phi(\Phi^2(\bot)) \\
 & \equiv & \Phi(x = y \vee R(x,y)) \\
 & \equiv & x = y \vee \exists z\ (R(x,z) \wedge (z = y \vee R(z,y))) \\
 & \equiv & x = y \vee R(x,y) \vee \exists z\ (R(x,z) \wedge R(z,y)) \\
\ldots & &
\end{array}$$

Thus $\Phi^i(\bot) \equiv \bigvee_{j<i} R^j(x,y)$ and therefore $\mu P(x).\Phi(P)$ describes the reflexive and ω-transitive closure of the relation R. ■

2.5 Modal Logics

Since we are mainly interested in the fixpoint characterization of modal logics, a short introduction to modal logics seems adequate. Evidently, this cannot be done here in every detail. The reader not experienced with modal logics is referred to [6] and [3].

It should be emphasized here that, for the sake of simplicity, we make the following restrictions:

- we consider one-argument modalities only
- we consider Kripke semantics only
- we do not consider first-order versions of modal logics
- we consider only normal modal logics.

However, the extension of our approach to cover the case of many-argument modalities as well as first-order quantifiers is straightforward. The method is even applicable to other kinds of semantics as well.

We consider *(propositional multi-) modal logic* with modalities $\{\Box_i\}_{i \in I}$, where I is an enumerable set. Let V be a set of propositional variables. The set of formulae $\mathcal{F}$ is then defined as the least set such that

- $V \subset \mathcal{F}$

- $\Phi, \Psi \in \mathcal{F}$ implies that $\overline{\Phi}, \Phi \vee \Psi, \Phi \wedge \Psi \in \mathcal{F}$ and for each $i \in I$, $\Box_i \Phi \in \mathcal{F}$.

The dual modalities $\Diamond_i$ and the boolean connectives $\supset, \equiv$ are defined as usual.

The semantics for modal logics is given by the Kripke semantics. A *Kripke frame* (frame, in short) is a pair $\langle W, \{\Re_i\}_{i \in I}\rangle$, where

- W is any non-empty set, called the set of *worlds*
- $\{\Re_i\}_{i \in I}$ is a set of binary relations defined on W, called the *accessibility relations.*

By a *Kripke model* we shall mean any triple $\langle \mathcal{K}, w, v\rangle$, where

- $\mathcal{K} = \langle W, \{\Re_i\}_{i \in I}\rangle$ is a Kripke frame
- $w \in W$ is a distinguished element of W, called the *actual* world
- $v : V \times W \longrightarrow \{\top, \bot\}$ is a mapping associating a truth value to a given propositional variable in a given world.

The *satisfiability relation* $\models$ is defined as follows:

- $\mathcal{K}, w, v \models p$ iff $v(p, w) = \top$, where $p \in V$ is a propositional variable
- $\mathcal{K}, w, v \models \overline{\Phi}$ iff not $\mathcal{K}, w, v \models \Phi$
- $\mathcal{K}, w, v \models \Phi \vee \Psi$ iff $\mathcal{K}, w, v \models \Phi$ or $\mathcal{K}, w, v \models \Psi$
- $\mathcal{K}, w, v \models \Phi \wedge \Psi$ iff $\mathcal{K}, w, v \models \Phi$ and $\mathcal{K}, w, v \models \Psi$
- $\mathcal{K}, w, v \models \Box_i \Phi$ iff for any w' such that $\Re_i(w, w')$, we have $\mathcal{K}, w', v \models \Phi$.

2.6 Semantics Based Translation

The semantics based translation we consider (sometimes called *standard translation* or *relational translation*) goes back at least to VanBenthem [20], where one can find the first systematic study of this notion (see also [10, 11, 12]). It consists mainly of a direct translation of the Kripke semantics for modal logics into first-order predicate logic. I.e. the domain under consideration is the set of worlds, the actual world is represented by a special world constant (or variable) and the accessibility relations $\Re_i$ by the binary predicate symbols R_i. Finally, for each propositional variable p a unary predicate symbol P is introduced which accepts a world (the actual world) as its only argument and is supposed to be evaluated to *true* if and only if the associated propositional variable is evaluated to *true* in the world given by the interpretation of the argument. Translation from modal logic formulae into predicate logic is then defined as follows:

$$
\begin{array}{lcl}
\lceil p \rceil_x & = & P(x) \\
\lceil \Box_i \Phi \rceil_x & = & \forall y\, R_i(x, y) \supset \lceil \Phi \rceil_y \\
\lceil \Diamond_i \Phi \rceil_x & = & \exists y\, R_i(x, y) \wedge \lceil \Phi \rceil_y
\end{array}
$$

where the translation for the classical connectives is obtained by the usual homomorphic extension to the above. Note that we sometimes use the modal operator symbol as the index of the corresponding accessibility relation symbol (e.g. $R_\Box$ for the accessibility relation of the $\Box$-operator).

This relational translation is sound and complete for the modal logic K. If we are interested in other modal logics, say *S4*, we have to add the correspondence axioms which characterize the modal logic under consideration (reflexivity and transitivity of R in case of *S4*). Thus the Correspondence Theory can be viewed as a discipline of finding such correspondence axioms.

We won't go into any further detail what the basics of the relational translation is concerned. Instead we provide with a small table of examples and encourage the reader to have a look at [3, 6, 9, 10, 11, 12] where this and other translation techniques are described in more detail.

Modal formula	Translated formula (with actual world x)
$\Box p \supset p$	$[\forall y\ R_\Box(x,y) \supset P(y)] \supset P(x)$
$\Box p \supset \Box\Box p$	$[\forall y\ R_\Box(x,y) \supset P(y)] \supset [\forall y\ (R_\Box(x,y) \supset \forall z(R_\Box(y,z) \supset P(z)))]$
$p \supset \Box\Diamond p$	$P(x) \supset [\forall y\ R_\Box(x,y) \supset \exists z\ (R_\Box(y,z) \wedge P(z))]$
$\Box(p \supset \bigcirc p) \supset (p \supset \Box p)$	$[\forall y\ (R_\Box(x,y) \supset (P(y) \supset \forall z\ (R_\bigcirc(y,z) \supset P(z))))] \supset$ $[P(x) \supset \forall u\ (R_\Box(x,u) \supset P(u))]$

Note that the above translation can easily be extended to the case of modal schemata, using second-order quantification as follows (cf. also [22]).

Modal schema	Translated schema
$\Box p$	$\forall P \forall x \forall y\ R_\Box(x,y) \supset P(y)$
$\Diamond p$	$\forall P \forall x \exists y\ R_\Box(x,y) \wedge P(y)$

The following table illustrates the translation of modal schemata.

Modal schema	Translated schema
$\Box p \supset p$	$\forall P \forall x(\forall y\ R_\Box(x,y) \supset P(y)) \supset P(x)$
$\Box p \supset \Box\Box p$	$\forall P \forall x(\forall y\ R_\Box(x,y) \supset P(y)) \supset \forall y(R_\Box(x,y) \supset \forall z(R_\Box(y,z) \supset P(z)))$
$p \supset \Box\Diamond p$	$\forall P \forall x\ P(x) \supset (\forall y\ R_\Box(x,y) \supset \exists z(R_\Box(y,z) \wedge P(z)))$
$\Box(p \supset \bigcirc p) \supset (p \supset \Box p)$	$\forall P \forall x[\forall y(R_\Box(x,y) \supset (P(y) \supset \forall z(R_\bigcirc(y,z) \supset P(z))))] \supset$ $[P(x) \supset \forall u(R_\Box(x,u) \supset P(u))]$

However, the above formulae are fairly useless from the point of view of automated theorem proving, as the set of second-order tautologies even with one second-order quantification is, in general, totally undecidable. In order to put our translation into good use, we should therefore try to eliminate the second-order quantifications and find some suitable, less complex conditions on R that could serve as a basis for modal logic theorem proving.

Note that the elimination technique we consider preserves logical equivalence of formulae. This means that the resulting formula is equivalent over all interpretations. Thus, whenever one considers semantics of modal logics which is based on some interpreted formalism (like neighborhood semantics), one has to work in some theory that approximates the formalism. However, we shall not consider the question of interpreted formalisms in this paper.

3 The Main Theorem

As mentioned earlier our aim is to find first-order correspondences for modal schemata. To this end we consider the relational translation approach described in the previous section (which results in second-order formulae) and apply our main theorem in order to obtain an equivalent first-order formula. This section is now meant to formulate and prove the main theorem that serves as a basis of the proposed approach.

Theorem 3.1
Assume that all occurrences of the predicate symbol P in the formula Ψ have only variables as arguments[1].

- if Φ and Ψ are negative w.r.t. P then the closure ordinal for $\Phi(\overline{P})$ is less than or equal to ω, and

$$\exists P \forall \bar{y}\, [P(\bar{y}) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})] \;\equiv\; \Psi[\overline{P} \leftarrow \nu \overline{P(\bar{y})}.\Phi(\overline{P})] \tag{9}$$

- if Φ and Ψ are positive w.r.t. P then the closure ordinal for $\Phi(P)$ is less than or equal to ω, and

$$\exists P \forall \bar{y} [\overline{P(\bar{y})} \vee \Phi(P)] \wedge [\Psi(P)] \;\equiv\; \Psi[P \leftarrow \nu P(\bar{y}).\Phi(P)] \tag{10}$$

where the above substitutions exchange the variables bound by fixpoint operators by the corresponding actual variables of the substituted predicate[2].

Proof
Let us prove the first part of the theorem, concerning equivalence (9) (the proof of (10) is then obvious).

We start by showing that the closure ordinal for $\Phi(\overline{P})$ is less or equal to ω. It suffices to prove that

$$\Phi(\overline{P} \leftarrow \bigvee_{i \in \omega} \Phi^i(\overline{P} \leftarrow \bot)) \;\equiv\; \bigvee_{i \in \omega} \Phi(\overline{P} \leftarrow \Phi^i(\overline{P} \leftarrow \bot)). \tag{11}$$

Without loss of generality we assume that Φ is in prenex and conjunctive normal form. By applying equivalences (5) and (6) one can transform this formula equivalently to the form

$$Q_1 \ldots Q_k \forall \bar{z} [(\overline{P(\bar{z})} \vee \Phi_1) \wedge \ldots \wedge (\overline{P(\bar{z})} \vee \Phi_r) \wedge \Phi'), \tag{12}$$

where $Q_1, \ldots, Q_k$ are first-order quantifiers and $\Phi_1, \ldots, \Phi_r, \Phi'$ are clauses containing no occurrences of $\overline{P}$. Formula (12) is equivalent to $Q_1 \ldots Q_k \forall \bar{z} [(\overline{P(\bar{z})} \vee (\Phi_1 \wedge \ldots \wedge \Phi_r)) \wedge \Phi']$. Next one can eliminate first-order existential quantifiers by applying equivalence (7). The formula takes the form $\exists \bar{f} \forall \bar{u} \forall \bar{z} [(\overline{P(\bar{z})} \vee (\Phi_1 \wedge \ldots \wedge \Phi_r)) \wedge \Phi']$, where $\bar{f}$ is a tuple of Skolem functions. One can now exchange the order of universal quantifiers and move $\forall \bar{u}$ inside. The resulting formula, equivalent to Φ, is of the form

$$\exists \bar{f} \forall \bar{z} [(\overline{P(\bar{z})} \vee (\forall \bar{u} \Phi_1 \wedge \ldots \wedge \forall \bar{u} \Phi_r)) \wedge \forall \bar{u} \Phi']. \tag{13}$$

The following equivalences show that (11) holds:

[1] Evidently, by applying the equivalence (5), every formula Ψ can easily be transformed to the required form.
[2] Observe that the assumption that P's in Ψ bind only variables is necessary here.

$$\begin{array}{l}
\Phi(\overline{P} \leftarrow \bigvee_{i\in\omega} \Phi^i(\overline{P} \leftarrow \bot)) \;\equiv \text{(by (13))}\\
\exists \bar{f}\forall\bar{z}[(\bigvee_{i\in\omega} \Phi^i(\overline{P} \leftarrow \bot) \vee (\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{u}\Phi'] \;\equiv\\
\exists \bar{f}[\forall\bar{z}(\bigvee_{i\in\omega} \Phi^i(\overline{P} \leftarrow \bot) \vee (\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{z}\forall\bar{u}\Phi'] \;\equiv\\
\qquad \text{(since } \bigvee_{i\in\omega} \Phi^i(\overline{P} \leftarrow \bot) \text{ does not contain free occurrences of } \bar{z})\\
\exists \bar{f}[(\bigvee_{i\in\omega} \Phi^i(\overline{P} \leftarrow \bot) \vee \forall\bar{z}(\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{z}\forall\bar{u}\Phi'] \;\equiv\\
\exists \bar{f}[\bigvee_{i\in\omega}(\Phi^i(\overline{P} \leftarrow \bot) \vee \forall\bar{z}(\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{z}\forall\bar{u}\Phi'] \;\equiv\\
\exists \bar{f} \bigvee_{i\in\omega}[(\Phi^i(\overline{P} \leftarrow \bot) \vee \forall\bar{z}(\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{z}\forall\bar{u}\Phi'] \;\equiv\\
\bigvee_{i\in\omega} \exists \bar{f}[(\Phi^i(\overline{P} \leftarrow \bot) \vee \forall\bar{z}(\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{z}\forall\bar{u}\Phi'] \;\equiv\\
\bigvee_{i\in\omega} \exists \bar{f}\forall\bar{z}[(\Phi^i(\overline{P} \leftarrow \bot) \vee (\forall\bar{u}\Phi_1 \wedge \ldots \wedge \forall\bar{u}\Phi_r)) \wedge \forall\bar{u}\Phi'] \;\equiv\\
\bigvee_{i\in\omega} \Phi(\overline{P} \leftarrow \Phi^i(\overline{P} \leftarrow \bot)).
\end{array}$$

Let us now show that equivalence (9) holds.
($\supset$) Assume that $\exists P\forall\bar{y}[P(\bar{y}) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})]$. Let P be such that the formula

$$\forall\bar{y}[P(\bar{y}) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})] \tag{14}$$

is satisfied. Observe that the first conjunct of formula (14) can be rewritten as $\forall\bar{y}[\overline{P(\bar{y})} \supset \Phi(\overline{P})]$. By assumption, Φ is negative w.r.t. P, thus positive (and monotone) w.r.t. $\overline{P}$. Obviously, $\overline{P(\bar{y})} \supset \top$. By monotonicity of Φ, $\Phi(\overline{P(\bar{y})}) \supset \Phi(\top)$, thus by the first conjunct of formula (14), $\overline{P(\bar{y})} \supset \Phi(\top)$. By induction one can show that for any $i \in \omega$, we have that $\overline{P(\bar{y})} \supset \Phi^i(\top)$. Thus also $\overline{P(\bar{y})} \supset \nu\overline{P(\bar{y})}.\Phi(\bar{P})$.

By assumption, P occurs only negatively in Ψ. Thus Ψ is positive (and monotone) w.r.t. $\overline{P}$. By the second conjunct of our formula (14), $\Psi(\overline{P(\bar{y})})$. This, together with the monotonicity of Ψ implies

$$\Psi[\overline{P} \leftarrow \nu\overline{P(\bar{y})}.\Phi(\overline{P})].$$

($\subset$) Assume that $\Psi[\overline{P} \leftarrow \nu\overline{P(\bar{y})}.\Phi(\overline{P})]$. In order to show that $\exists P\forall\bar{y}[P(\bar{y}) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})]$ we define $\overline{P(\bar{y})}$ to be $\nu\overline{P(\bar{y})}.\Phi(\overline{P})$, i.e.

$$P(\bar{y}) \;\equiv\; \mu P(\bar{y}).\overline{\Phi}(P). \tag{15}$$

By (15) we have that $P(\bar{y})$ is a fixpoint of $\overline{\Phi}(P)$, i.e. $P(\bar{y}) \;\equiv\; \overline{\Phi}(P)$. Thus

$$\exists P\forall\bar{y}\;[P(\bar{y}) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})]$$

is equivalent to

$$\exists P\forall\bar{y}\;[\overline{\Phi}(P) \vee \Phi(\overline{P})] \wedge [\Psi(\overline{P})],$$

i.e. to $\exists P[\Psi(\overline{P})]$, which holds by our assumption. ■

The following proposition shows that the above theorem subsumes the lemma of Ackermann.

Proposition 3.2
Theorem 3.1 subsumes Lemma 2.2.

Proof

Let us prove that equivalence (9) implies equivalence (3) (the proof that (10) implies (4) is similar).

Let assumptions of Lemma 2.2 be satisfied. In particular, $\Phi(\overline{P})$ of equivalence (9), contains no occurrences of P and $\overline{P}$. Thus

$$\nu\overline{P(\bar{y})}.\Phi(\overline{P}) \;\equiv\; \Phi(\overline{P}) \;\equiv\; \Phi \tag{16}$$

i.e. the righthand side of (9) reduces to $\Psi[\overline{P} \leftarrow \Phi]$, just as required in Lemma 2.2. ∎

Observe that equivalence (16) given in the proof of the above proposition shows that the fixpoint formulas, resulting from Theorem 3.1 can be reduced to first-order formulas whenever assumptions of the Ackermann lemma are satisfied.

In order to transform formulae into the form required in Theorem 3.1 we apply well-known tautologies of the classical first-order logic, in particular (5), (6) and (7) of Section 2.3.

4 Applications

Now let us come to some application examples of our approach. The strategy is always essentially the same for all of the examples. I.e. first the problem has to be transformed into the form $\exists P\ \Phi$ where P is a predicate symbol (the one we try to eliminate) and Φ is a first-order formula. Modal schemata are not initially of this form. We therefore have to translate the schema with the help of the relational translation approach and to negate its result (the translation results in a universal second-order quantification). After that the formula has to be transformed into the form required for the main theorem to be applicable. One can construct an algorithm based on equivalences presented in Section 2.3 that performs the required transformation. An algorithm that can easily be adapted here is given in [18] and, in a stronger form, in [4]. The application of Theorem 3.1 results in a fixpoint formula which has to be negated again. Finally some simplification transformations are performed if possible and we are done.

Example 4.1
Let us first have another look at Example 2.1 where SCAN ran into a loop, i.e. consider again the formula (2)

$$\forall P\ [(\forall x\ P(x) \supset \exists y\ (P(y) \wedge Q(x))) \supset \forall x\ \overline{P(x)}].$$

Negated and transformed into the form required in Theorem 3.1 —
$\exists x \exists P \forall y\ (\overline{P(y)} \vee \exists z\ (P(z) \wedge Q(y))) \wedge P(x)$
After application of Theorem 3.1 —
$\exists x\ \nu P(x).[\exists z\ P(z) \wedge Q(x)]$, i.e. $\exists x\ \nu P(x).[\exists z\ P(z)] \wedge Q(x)$
Simplified[3] — $\exists x\ Q(x)$
Unnegated — $\forall x\ \overline{Q(x)}$. ∎

The following example illustrates the use of Theorem 3.1 in case of the second-order induction axiom.

Example 4.2
Consider the second-order induction, where $S(x, y)$ means that y is a successor of x:

$$\forall P\ [P(0) \wedge \forall x \forall y\ ((P(x) \wedge S(x, y)) \supset P(y))] \supset \forall z\ P(z) \tag{17}$$

Let us now transform the formula into the form required in Theorem 3.1 and then apply the theorem:

[3]Using the equivalence $\nu P(x).[\exists z\ P(z)] \equiv \top$

Negated —

$$\exists P[P(0) \wedge \forall x \forall y((P(x) \wedge S(x,y)) \supset P(y))] \wedge \exists z \overline{P(z)}$$

Transformed into the conjunctive normal form —

$$\exists z \exists P \forall y[(y \neq 0 \vee P(y)) \wedge \forall x(\overline{P(x)} \vee \overline{S(x,y)} \vee P(y))] \wedge \overline{P(z)}$$

Transformed into the form required in Theorem 3.1 —

$$\exists z \exists P \forall y[P(y) \vee (y \neq 0 \wedge \forall x(\overline{S(x,y)} \vee \overline{P(x)}))] \wedge \overline{P(z)}$$

After application of Theorem 3.1 —

$$\exists z\ \nu \overline{P(z)}.(z \neq 0 \wedge \forall x[\overline{S(x,z)} \vee \overline{P(x)})]$$

Unnegated —

$$\forall z\ \mu P(z).[z = 0 \vee \exists x(S(x,z) \wedge P(x))].$$

By Theorem 3.1 we have that the above formula is equivalent to the following infinite disjunction:

$$\forall z \bigvee_{i \in \omega} [z = 0 \vee \exists x(S(x,z) \wedge P(x))]^i(\bot).$$

Denote by $\Phi(\Psi(z))$ the formula $z = 0 \vee \exists x(S(x,z) \wedge \Psi(x))$. Now

$$\begin{aligned}
\Phi^0(\bot) &\equiv \bot \\
\Phi^1(\bot) &\equiv [z = 0 \vee \exists x(S(x,z) \wedge \bot)] \equiv [z = 0 \vee \bot] \equiv [z = 0] \\
\Phi^2(\bot) &\equiv \Phi(\Phi^1(\bot)) \equiv [z = 0 \vee \exists x(S(x,z) \wedge x = 0)] \equiv [z = 0 \vee S(0,z)] \\
\Phi^3(\bot) &\equiv \Phi(\Phi^2(\bot)) \equiv [z = 0 \vee \exists x(S(x,z) \wedge (x = 0 \vee S(0,x)))] \equiv \\
&\quad [z = 0 \vee S(0,z) \vee S^2(0,z)] \\
&\ldots \\
\Phi^i(\bot) &\equiv [z = 0 \vee S(0,z) \vee \ldots \vee S^{i-1}(0,z)].
\end{aligned}$$

Thus, by a simple calculation we obtain the well-known, but not trivial fact that the second-order induction (17) is equivalent to

$$\forall z \bigvee_{i \in \omega} [S^i(0,z)],$$

i.e. "every natural number is obtained from 0 by a finite number of applications of the successor relation". In fact, the other Peano axioms say that the successor relation is a function, etc. ∎

Example 4.3

Assume that R is transitive and consider the formula

$$\forall P\ [\forall x\ \overline{P(x)} \vee \forall y\ P(y) \vee \exists x \exists y\ (P(x) \wedge R(x,y) \wedge \overline{P(y)})]. \tag{18}$$

Note that formula (18) has a first-order equivalent over transitive frames, namely (19). However SCAN loops and the method based on the Ackermann lemma fails with (18) given as an input. Hence let us apply Theorem 3.1.

Negated — $\exists P[\exists x\ P(x) \wedge \exists y\ \overline{P(y)} \wedge \forall x \forall y\ (\overline{P(x)} \vee \overline{R(x,y)} \vee P(y))]$

Transformed into an equivalent form —

$$\exists x \exists y \exists P\ [\forall w\ (P(w) \vee \forall u(\overline{P(u)} \vee \overline{R(u,w)})) \wedge \forall w\ (P(w) \vee x \neq w) \wedge \overline{P(y)}]$$

Transformed into the form required in Theorem 3.1 —

$$\exists x \exists y \exists P[\forall w\ (P(w) \vee (x \neq w \wedge \forall u\ (\overline{R(u,w)} \vee \overline{P(u)})))] \wedge \overline{P(y)}$$

After application of Theorem 3.1 —

$$\exists x \exists y\ \nu \overline{P(y)}.[x \neq y \wedge \forall u\ (\overline{R(u,y)} \vee \overline{P(u)})]$$

Unnegated — $\forall x \forall y\ \mu P(y).[x = y \vee \exists u\ (R(u,y) \wedge P(u))]$.

Denote by $\Phi(P(y))$ the formula $x = y \vee \exists u\ (R(u,y) \wedge P(u))$. Then

$$
\begin{aligned}
&\Phi^0(\bot) \ \equiv\ \bot \\
&\Phi^1(\bot) \ \equiv\ x = y \vee \exists u(R(u,y) \wedge \bot) \ \equiv\ x = y \\
&\Phi^2(\bot) \ \equiv\ \Phi(\Phi^1(\bot)) \ \equiv\ x = y \vee \exists u(R(u,y) \wedge x = u) \ \equiv\ x = y \vee R(x,y) \\
&\Phi^3(\bot) \ \equiv\ \Phi(\Phi^2(\bot)) \ \equiv\ x = y \vee \exists u(R(u,y) \wedge (x = u \vee R(x,u))) \ \equiv \\
&\quad x = y \vee R(x,y) \vee \exists u(R(u,y) \wedge R(x,u))
\end{aligned}
$$

Evidently, $\exists u\ (R(u,y) \wedge R(x,u))$ together with the transitivity of R implies $R(x,y)$. It therefore follows that

$$\Phi^3(\bot) \ \equiv\ x = y \vee R(x,y) \ \equiv\ \Phi^2(\bot)$$

Thus $\Phi^i(\bot) \ \equiv\ x = y \vee R(x,y)$ for all $i \geq 2$ and therefore we have (under the assumption that R is transitive) that formula (18) is equivalent to

$$\forall x \forall y\ (x = y \vee R(x,y)) \tag{19}$$

i.e. it represents the reflexive closure of R. ■

So far we considered merely some general second-order quantification elimination problems. Let us now show some correspondence-theoretical applications of Theorem 3.1.

Example 4.4
Consider the Löb Axiom

$$\Box(\Box P \supset P)\ \supset\ \Box P \tag{20}$$

Translated —
$$\forall P \forall x\ [\forall y\ (R(x,y) \supset \forall z\ (R(y,z) \supset P(z)) \supset P(y))] \supset \forall u\ (R(x,u) \supset P(u))$$
After negation and elimination of $\supset$ —
$$\exists P \exists x\ [\forall y\ (\overline{R(x,y)} \vee \exists z\ (R(y,z) \wedge \overline{P(z)} \vee P(y)))] \wedge \exists u\ (R(x,u) \wedge \overline{P(u)})$$
Transformed into the form required in Theorem 3.1 —
$$\exists x \exists P\ [\forall y\ (P(y) \vee \overline{R(x,y)} \vee \exists z\ (R(y,z) \wedge \overline{P(z)}))] \wedge \exists u\ (R(x,u) \wedge \overline{P(u)})$$
After application of Theorem 3.1 —
$$\exists x \exists u\ R(x,u) \wedge \nu \overline{P(u)}.[\overline{R(x,u)} \vee \exists z\ (R(u,z) \wedge \overline{P(z)})]$$
Unnegated —
$$\forall x \forall u\ R(x,u) \supset \mu P(u).[R(x,u) \wedge \forall z\ (R(u,z) \supset P(z))]$$

Denote by $\Phi(\Psi(u))$ the formula $R(x,u) \wedge \forall z\ (R(u,z) \supset \Psi(z))$. Now

$$
\begin{aligned}
&\Phi^0(\bot) \ \equiv\ \bot \\
&\Phi^1(\bot) \ \equiv\ R(x,u) \wedge \forall z\ \overline{R(u,z)} \\
&\Phi^2(\bot) \ \equiv\ \Phi(\Phi(\bot)) \ \equiv\ \ \Phi(R(x,u) \wedge \forall z\ \overline{R(u,z)}) \equiv \\
&\quad R(x,u) \wedge \forall z\ (R(u,z) \supset (R(x,z) \wedge \forall z_1\ \overline{R(z,z_1)}))) \\
&\Phi^3(\bot) \ \equiv\ \Phi(\Phi^2(\bot)) \ \equiv \\
&\quad \Phi(R(x,u) \wedge \forall z\ (R(u,z) \supset (R(x,z) \wedge \forall z_1\ \overline{R(z,z_1)}) \equiv \\
&\quad R(x,u) \wedge \forall z\ (R(u,z) \supset R(x,z) \wedge \forall z_2\ (R(z,z_2) \supset (R(x,z_2) \wedge \forall z_1\ \overline{R(z_2,z_1)}))) \\
&\ldots
\end{aligned}
$$

Thus the Löb axiom (20) is equivalent to

$$\forall x \forall u\, R(x,u) \supset \bigvee_{i \in \omega} \Phi^i(\bot),$$

which expresses that the relation R is transitive and reverse well-founded. Transitivity of R can be seen by unfolding the fixpoint two times and the reverse well-foundedness of R then follows from the simplification of $\forall x \forall u\ R(x,u) \supset \mu P(u).[R(x,u) \wedge \forall z\ (R(u,z) \supset P(z))]$ to $\forall x \forall u\, R(x,u) \supset \mu P(u).[\forall z\, R(u,z) \supset P(z)]$ under the transitivity assumption. ∎

One of the open questions listed in [14], p. 168, is formulated as follows:

> "The Gabbay-Ohlbach-Simmons algorithm was unable to deal with the Löb formula $\Box(\Box p \supset p) \supset \Box p$, despite the fact that it does have a p-free equivalent (namely well-foundedness). What further functions need to be assumed present to make an extension of the algorithm find this equivalent?"

Example 4.4 shows that our approach is powerful enough to deal with Löb-like formulas, too. This provides us with yet another argument supporting the use of fixpoint calculus in the context of Correspondence Theory.

Next we consider a modal schema which is not validated by any frame. It is interesting in so far as SCAN is again not able prove this fact. Similarly, the application of Lemma 2.2 is not possible here.

Example 4.5
Consider the axiom schema

$$\Box(\Box P \supset P) \supset P \tag{21}$$

Translated —
$\forall P \forall x\, [\forall y(R(x,y) \supset (\forall z\ (R(y,z) \supset P(z)) \supset P(y))) \supset P(x)]$
Negated and transformed into the form required in Theorem 3.1 —
$\exists x \exists P\ \forall y\ [P(y) \vee \overline{R(x,y)} \vee \exists z\ (R(y,z) \wedge \overline{P(z)})] \wedge \overline{P(x)}$
After application of Theorem 3.1 —
$\exists x\ \nu \overline{P(x)}.[\overline{R(x,x)} \vee \exists z\ (R(x,z) \wedge \overline{P(z)})]$
Unnegated —
$\forall x\ \mu P(x).[R(x,x) \wedge \forall z\ (\overline{R(x,z)} \vee P(z))]$.

Denote by $\Phi(P(x))$ the formula $R(x,x) \wedge \forall z\ (\overline{R(x,z)} \vee P(z))$. Now

$$\begin{aligned} \Phi^0(\bot) &\equiv \bot \\ \Phi^1(\bot) &\equiv R(x,x) \wedge \forall z\ \overline{R(x,z)} \ \equiv\ \bot \end{aligned}$$

Thus $\Phi^i(\bot)\ \equiv\ \bot$ for all i, hence the schema (21) is equivalent to $\bot$, i.e. no frame validates this schema. ∎

Example 4.6
Consider the following axiom schema:

$$\Box(P \supset \Diamond P)\ \supset\ (\Diamond P \supset \Box\Diamond P) \tag{22}$$

which is sometimes called the *modified Löb Axiom* (see [22])[4].

Translated —
$\forall P \forall x\ [\forall y\ (R(x,y) \supset (P(y) \supset \exists z\ (R(y,z) \wedge P(z)))] \supset$
$[\exists u\ (R(x,u) \wedge P(u)) \supset \forall v\ (R(x,v) \supset \exists w\ (R(v,w) \wedge P(w)))]$
Negated —
$\exists x\ \exists P\ [\forall y\ (R(x,y) \supset (P(y) \supset \exists z\ (R(y,z) \wedge P(z)))] \wedge$
$\exists u\ (R(x,u) \wedge P(u)) \wedge \exists v\ (R(x,v) \wedge \forall w\ (\overline{R(v,w)} \vee \overline{P(w)}))$
Transformed into an equivalent form —
$\exists x \exists u \exists v\ \exists P\ [\forall y\ (\overline{R(x,y)} \vee \overline{P(y)} \vee \exists z\ (R(y,z) \wedge P(z))) \wedge$
$(R(x,u) \wedge P(u)) \wedge (R(x,v) \wedge \forall w\ (\overline{R(v,w)} \vee \overline{P(w)}))]$
Transformed into the form required in Theorem 3.1 —
$\exists x \exists u \exists v\ \exists P\ \forall y\ [\overline{P(y)} \vee (\overline{R(v,y)} \wedge (\overline{R(x,y)} \vee \exists z\ (R(y,z) \wedge P(z))))] \wedge$
$[R(x,u) \wedge P(u) \wedge R(x,v)]$
After application of Theorem 3.1 —
$\exists x \exists u \exists v\ R(x,u) \wedge R(x,v) \wedge \nu P(u).[\overline{R(v,u)} \wedge (\overline{R(x,u)} \vee \exists z\ (R(u,z) \wedge P(z)))]$
Unnegated —
$\forall x \forall u \forall v\ R(x,u) \wedge R(x,v) \supset \mu \overline{P(u)}.[R(v,u) \vee (R(x,u) \wedge \forall z\ (R(u,z) \supset \overline{P(z)}))]$

It is still not easy to provide with a nice intuition about this property. Its strong relation to the Löb axiom is pretty evident, however. As Arthur Prior mentions in [13], it allows to prove Dummet's axiom of discreteness under the Diodorian Modality. Also it serves as a schema for the tense logic definability of the natural numbers under the $<$-relation (see [22]). ■

Example 4.7
Consider the temporal logic formula

$$\Box(p \supset \bigcirc p)\ \supset\ (p \supset \Box p). \tag{23}$$

where $\Box$ should be interpreted as *always* or *henceforth* and $\bigcirc$ as *at the next moment of time.*

Translated —
$\forall P\ \forall x\ [\forall y\ (R_{\Box}(x,y) \supset (P(y) \supset \forall z\ (R_{\bigcirc}(y,z) \supset P(z))))] \supset$
$[P(x) \supset \forall u\ (R_{\Box}(x,u) \supset P(u))]$
Negated —
$\exists x\ \exists P\ [\forall y\ (R_{\Box}(x,y) \supset (P(y) \supset \forall z\ (R_{\bigcirc}(y,z) \supset P(z))))] \wedge$
$[P(x) \wedge \exists u\ (R_{\Box}(x,u) \wedge \overline{P(u)})]$
Transformed into the conjunctive normal form —
$\exists x \exists u\ \exists P\ \forall y \forall z\ (P(z) \vee \overline{R_{\Box}(x,y)} \vee \overline{R_{\bigcirc}(y,z)} \vee \overline{P(y)}) \wedge$
$\forall z\ (P(z) \vee x \neq z) \wedge R_{\Box}(x,u) \wedge \overline{P(u)}$
Transformed into the form required in the Theorem 3.1 —
$\exists x \exists u\ \exists P\ \forall z\ [P(z) \vee (x \neq z \wedge \forall y\ (\overline{R_{\Box}(x,y)} \vee \overline{R_{\bigcirc}(y,z)} \vee \overline{P(y)}))] \wedge R_{\Box}(x,u) \wedge \overline{P(u)}$
After application of Theorem 3.1 —
$\exists x \exists u\ R_{\Box}(x,u) \wedge \nu \overline{P(u)}.(x \neq u \wedge \forall y\ (\overline{R_{\Box}(x,y)} \vee \overline{R_{\bigcirc}(y,u)} \vee \overline{P(y)}))]$
Unnegated — $\forall x \forall u\ R_{\Box}(x,u) \supset \mu P(u).(x = u \vee \exists y\ (R_{\Box}(x,y) \wedge R_{\bigcirc}(y,u) \wedge P(y)).$

[4]In a temporal interpretation this schema is often examined in its equivalent form $(\Diamond P \wedge \Diamond \Box \overline{P}) \supset \Diamond(P \wedge \Box \overline{P})$, i.e. *if P has not yet stopped being true but will do so eventually then there will be a final moment of P's truth.*

Thus formula (23) is equivalent to the following one:

$$\forall x \forall u\ R_{\Box}(x,u) \supset [x = u \vee \bigvee_{i\in\omega} \exists y_0 \ldots \exists y_i (R_{\bigcirc}(x,y_0) \wedge \ldots \wedge R_{\bigcirc}(y_{i-1},y_i) \wedge R_{\bigcirc}(y_i,u))],$$

or to the even simpler looking

$$\forall x \forall u\ R_{\Box}(x,u) \supset \bigvee_{i\in\omega} R^i_{\bigcirc}(x,u)$$

i.e. (23) states that $R_{\Box}$ is contained in the reflexive and ω-transitive closure of $R_{\bigcirc}$, a property which is not expressible by means of classical first-order logic but expressible by means of fixpoint logic. ■

5 Comparison of Approaches

The following list indicates the relation of Theorem 3.1 to other approaches to automatizing Correspondence Theory:

- as shown in Proposition 3.2, Theorem 3.1 subsumes the Ackermann lemma. As shown in Examples 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, this subsumption is strict. Observe that Examples 4.3 and 4.5 show formulas for which Ackermann lemma fails, but there is a first-order equivalent of those formulas. Thus Theorem 3.1 substantially extends the approach presented in [1, 18, 19] and the DLS algorithm of [4]
- as shown in Examples 4.3, 4.5 and Example 2.1 together with 4.1, Theorem 3.1 reduces formulas that are not reducible by SCAN algorithm of [5]. Examples 4.2, 4.4, 4.6, 4.7 show that Theorem 3.1 gives as a result a fixpoint formula, while SCAN loops. Thus our approach extends SCAN algorithm. However, we conjecture that SCAN is not subsumed by Theorem 3.1
- Example 4.5 shows a reduction of a formula that cannot be reduced by Sahlquist theorem (see [22]; a generalized definition can be found in [14]). Examples 4.4, 4.6 and 4.7 show a reduction to fixpoint logic of non-Sahlquist formulas. In fact, it can be easily observed that Sahlquist formulas, when given as input to our theorem, are in separated forms, i.e. positive and negative occurrences of the eliminated predicate are separated. Theorem 3.1 reduces in this case to the Ackermann lemma. Now it can easily be observed that formulas that are type 4 distributive (in the terminology of [14]) are reducible by Ackermann lemma. Sahlquist formulas are then Ackermann-reducible formulas. Thus our theorem extends the Sahlquist theorem. The same applies to the Sahlquist-VanBenthem algorithm (see [14]).

6 Conclusions

In this paper we presented an automated technique of eliminating second-order quantifiers from formulae. The technique we consider substantially extends known approaches. The resulting formulae are in general first-order or fixpoint formulae that are relatively easy to handle and to understand.

Although we mainly concentrated on the modal correspondence theory, our fixpoint approach is much more general. In particular it can be applied in the following important areas:

- modal and second-order theorem proving, since fixpoint calculus have a nice proof theory (cf. [8, 16, 17]). The Park rule

$$\Psi(I) \supset I \ \vdash \ \mu P.\Psi(P) \supset I$$

  and its ν-formulation

$$I \supset \Psi(I) \ \vdash \ I \supset \nu P.\Psi(P)$$

  are particularly useful here, since the formulae that result from application of Theorem 3.1 are monotone w.r.t. predicates substituted by fixpoint formulae

- modal and second-order model checking over finite domains, since model checking in fixpoint case can be done in polynomial time (cf. [7]), while second-order model checking over finite domains is NP-hard (by Fagin's theorem [7])

- mechanizing commonsense reasoning, e.g. based on circumscription (recall, that circumscription introduces second-order quantification).

It is worth emphasizing here that the formulae of the form required in Theorem 3.1 are, in general, not reducible to first-order formulae. E.g. the second-order induction considered in Example 4.2 has no first-order equivalent. Similarly, most of the modal schemata considered in this paper are not first-order definable. Thus the extension of classical first-order logic by fixpoint formulae seems to be, in a sense, minimal.

Also it should be mentioned that this approach does not always lead to either a first-order formula or some fixpoint formula (fails, e.g. in case of McKinsey axiom - see [18] in case of Ackermann lemma; Theorem 3.1 does not help either). Recall that it is sometimes necessary to apply an equivalence preserving skolemization in order to be able to transform the input formula into the form required for our main theorem. Such a skolemization evidently introduces a quantification over functions and therefore the final result might not be a first-order formula even if the generated fixpoint itself causes no further problems. However, it is often possible to simplify such a quantification over functions such that the final result is again of first-order.

References

[1] W. Ackermann: *Untersuchungen über das Eliminationsproblem der mathematischen Logik*, Mathematische Annalen, 110 (1935), 390–413.

[2] R. A. Bull, K. Segerberg: *Basic Modal Logic*, in: Handbook of Philosophical Logic, Vol. 2 (D. Gabbay and F. Guenthner, eds.), D. Reidel Pub. Co. (1984), 1–88.

[3] B. Chellas: *Modal Logic: An Introduction*, Cambridge University Press (1980).

[4] P. Doherty, W. Łukaszewicz & A. Szałas: *Computing Circumscription Revisited: A Reduction Algorithm* (1994), Technical Report, Linköping University, IDA, LiTh-IDA-R-94-42, Sweden. A preliminary report published in Proceedings of the IJCAI'95, Morgan Kaufmann (1995), 1502-1508. Full version to appear in Journal of Automated Reasoning.

[5] D. Gabbay & H. J. Ohlbach: *Quantifier elimination in second–order predicate logic*, in: Proceedings of the KR'92, Principles of Knowledge Representation and Reasoning, Morgan Kaufmann (1992), 425–436.

[6] G. E. Hughes & M. J. Cresswell: *An Introduction to Modal Logic*, Routledge (1989).

[7] N. Immerman: *Descriptive and Computational Complexity*, in: Proceedings of Symposia in Applied Mathematics 38 (1989), 75–91.

[8] D. Kozen: *Results on the Propositional μ-calculus*, Theoretical Computer Science 27 (1983), 333–354.

[9] R. Moore: *Reasoning About Knowledge and Action*, Ph.D. Thesis, MIT, Cambridge (1980).

[10] A. Nonnengart: *First-Order Modal Logic Theorem Proving and Functional Simulation*, in: Proceedings of the IJCAI'93, Morgan Kaufmann (1993), 80–85.

[11] A. Nonnengart: *How to Use Modalities and Sorts in Prolog*, in: Proceedings of the JELIA'94, LNAI 838, Springer Verlag (1994), 365–378.

[12] H. J. Ohlbach: *Semantics-Based Translation Methods for Modal Logics*, Journal of Logic and Computation, 5 (1991), 691–746.

[13] A. Prior: *Past, Present and Future*, Clarendon Press, Oxford (1967).

[14] M. de Rijke: *Extending Modal Logic*, Ph.D. Thesis, Institute for Logic, Language and Computation, University of Amsterdam (1993).

[15] H. Simmons: *The Monotonous Elimination of Predicate Variables*, Journal of Logic and Computation, 4 (1994), 23–68.

[16] A. Szałas: *On Strictly Arithmetical Completeness in Logics of Programs*, Theoretical Computer Science 79 (1991), 341–355.

[17] A. Szałas: *Axiomatizing Fixpoint Logics*, Information Processing Letters, 41 (1992), 175–180.

[18] A. Szałas: *On the Correspondence Between Modal and Classical Logic: an Automated Approach*, Journal of Logic and Computation, 3 (1993), 605–620.

[19] A. Szałas: *On an Automated Translation of Modal Proof Rules into Formulas of the Classical Logic*, Journal of Applied Non-Classical Logics, 4 (1994), 119–128.

[20] J. van Benthem: *Modal Correspondence Theory*, Ph.D. Thesis, Mathematisch Institut & Institut voor Grondslagenonderzoek, University of Amsterdam (1976).

[21] J. van Benthem: *Modal Logic and Classical Logic*, Bibliopolis, Napoli (1983).

[22] J. van Benthem: *Correspondence Theory*, in: Handbook of Philosophical Logic, Vol. 2 (D. Gabbay and F. Guenthner, eds.), D. Reidel Pub. Co. (1984), 167–247.