Second-Order Quantifier Elimination on Relational Monadic Formulas – A Basic Method and Some Less Expected Applications

Christoph Wernhard

Technische Universität Dresden

Abstract. For relational monadic formulas (the Löwenheim class) second-order quantifier elimination, which is closely related to computation of uniform interpolants, forgetting and projection, always succeeds. The decidability proof for this class by Behmann from 1922 explicitly proceeds by elimination with equivalence preserving formula rewriting. We reconstruct Behmann's method, relate it to the modern DLS elimination algorithm and show some applications where the essential monadicity becomes apparent only at second sight. In particular, deciding $\mathcal{ALCOQH}$ knowledge bases, elimination in DL-Lite knowledge bases, and the justification of the success of elimination methods for Sahlqvist formulas.

1 Introduction

A procedure for *second-order quantifier elimination* takes a second-order formula as input and yields an equivalent first-order formula in which the quantified predicates do no longer occur, and in which also no new predicates, constants or free variables are introduced. Obviously, on the basis of classical first-order logic this is not possible in general. Closely related are *uniform interpolation* and *projection*, where the predicates that are *not* eliminated are made explicit, *forgetting* where elimination of particular ground atoms is possible, and *literal forgetting* which can apply to just the predicate occurrences with positive or negative polarity. These variants are often also based on a syntactic view, characterized in terms of the *set of consequences* of the result formula instead of equivalence.

Second-order quantifier elimination and its variants have many applications in knowledge processing, including ontology reuse, ontology analysis, logical difference, information hiding, computation of circumscription, abduction in logic programming and view-based query processing [20,32,31,15,47,48]. It thus seems useful to consider as a requirement of knowledge representation languages in addition to decidability also "eliminability", that elimination of symbols succeeds. If eliminating all symbols yields *true* or *false*, this implies decidability.

The two main approaches for second-order quantifier elimination with respect to first-order logic are resolvent generation [19,18] and the *direct methods*, where formulas are rewritten into a shape that immediately allows elimination according to schematic equivalences such as Ackermann's Lemma [1,15,18]. In particular for modal and description logic some dedicated elimination methods have been presented in explicit relation to these two approaches, e.g., [27,12,41], while several others only in context of the considered special logic, e.g., [24,45,31].

The general characterization of formula classes that allow successful elimination is not yet thoroughly researched. Some of the mentioned methods and investigations such as [11] give indications. Further subtle questions arise if not just the symbols in the result but also further properties – such as belonging again to the input class – are taken into consideration.

In this paper, we approach that scenario from the viewpoint of a working hypothesis that might be stated as "many applications are actually instances of a modest subclass of first-order logic that allows elimination and is characterized by a general criterion." Consequences in perspective would be that the reducibility to the modest class provides explanations for the success of elimination, that possibly interesting boundaries come to light when a feature is really inexpressible in the modest class, that results apply in the context of first-order logic as a general framework with many well developed techniques and allowing to embed other logics, and that a modest class could facilitate efficient implementation.

A look back into history highlights the class of relational monadic formulas as candidate of such a "modest class." For its variants, we use the following symbols: MON is the class of relational monadic formulas (also called Löwenheim class), that is, the class of first-order formulas with nullary and unary predicates, with individual constants but no other functions and without equality. $\mathsf{MON}_=$ is MON with equality. QMON and $\mathsf{QMON}_=$ are MON and $\mathsf{MON}_=$, resp., extended by second-order quantification upon predicates.

All of these classes are decidable. $\mathsf{QMON}_=$ admits second-order quantifier elimination, that is, there is an effective method to compute for a given $\mathsf{QMON}_=$ formula F an equivalent $\mathsf{MON}_=$ formula F' in which all predicates are unquantified predicates in F, as well as all constants and free variables are also in F. In this sense $\mathsf{MON}_=$ is closed under second-order quantifier elimination, which does not hold for MON, since elimination applied to a QMON formula might introduce equality. These results have been obtained rather early by Löwenheim [33], Skolem [43] and Behmann [5]. The first documented use of *Entscheidungsproblem* actually seems to be the registration of a talk by Behmann in 1921 [51,34]. We focus here on Behmann's decision procedure for several reasons: It aims at practical application, operating in a way that appears rather modern by equivalence preserving formula rewriting. It provides a link between the decision problem and elimination by the reduction of deciding satisfiability to successive elimination of all predicates. In addition, motivated by earlier works of Ernst Schröder, the application to elimination problems on their own has been considered.

Behmann's elimination procedure can be seen as an early instance of the direct methods, where formulas are rewritten until subformulas with predicate quantification match an elimination schema. In the case of DLS [15] this schema is Ackermann's Lemma, a side result of [1]. Actually, Ackermann acknowledged that Behmann's paper [5] was at its time the impetus for him to investigate the elimination problem in depth (letter to Behmann, 29 Oct 1934, [6]). In modern expositions of second-order quantifier elimination, e.g., [18], Behmann's contributions have so far been largely overlooked with exception of historic references [14,41]. A comprehensive summary of the contributions is given in [49].

The rest of the paper is structured as follows: After fixing notational conventions, we present a restoration of Behmann's elimination method (Sect. 2) and properties of second-order logic that will be useful in the sequel (Sect. 3).

In Sect. 4 description logics are considered: It is shown that decidability of $\mathcal{ALCOQH}$ knowledge bases can be polynomially reduced to decidability of relational monadic formulas. With respect to elimination problems, related mappings are possible for description logics of the DL-Lite family. Some issues and subtleties that arise for elimination via such mappings are discussed. In Sect. 5 direct methods with Ackermann's Lemma are related to monadic techniques. A possibility to improve DLS becomes apparent and it is shown that a condition related to monadicity can serve as explanation for the success of methods based on Ackermann's Lemma. The success of the Sahlqvist-van Benthem substitution method and of DLS for computing first-order correspondence properties of Sahlqvist formulas can be attributed to that property. Finally, related work is discussed in Sect. 6 and concluding remarks are provided in Sect. 7. This report is an extended version of [50].

Notational Conventions. We consider formulas constructed from atoms, constant operators $\top$, $\bot$, the unary operator $\neg$, binary operators $\wedge, \vee$ and quantifiers $\forall, \exists$ with their usual meaning. The scope of quantifiers is understood as extending as far to the right as possible. A subformula occurrence has in a given formula positive (negative) *polarity* if it is in the scope of an even (odd) number of negations. Negated equality $\neq$, further binary operators $\rightarrow, \leftarrow, \leftrightarrow$, as well as n-ary versions of $\wedge$ and $\vee$ can be understood as meta-level notation. The scope of n-ary operators in prefix notation is the immediate subformula to the right. *Counting quantifiers* $\exists^{\geq n}$, where $n \geq 1$, express existence of at least n individuals. Two alternate expansions into first-order logic are as follows: Let $F[x]$ be a formula in which x possibly occurs free, let $x_1, \ldots, x_n$ be fresh variables, and let $F[x_i]$ denote $F[x]$ with the free occurrences of x replaced by x_i. It then holds that $\exists^{\geq n} x\, F[x] \equiv \exists x_1 \ldots \exists x_n \bigwedge_{1 \leq i \leq n} F[x_i] \wedge \bigwedge_{i<j\leq n} x_i \neq x_j \equiv \forall x_1 \ldots \forall x_{n-1} \exists x\, F[x] \wedge \bigwedge_{1 \leq i < n} x \neq x_i$. A *Boolean combination of basic formulas* is a formula obtained from certain basic formulas and the operators $\top, \bot, \neg, \wedge, \vee$.

2 Behmann's Elimination Method

The core property shown in [5] can be stated as follows:

Proposition 1 (Predicate Elimination for $\mathsf{MON}_=$). *There is an effective method to compute from a given predicate p and $\mathsf{MON}_=$ formula F a formula F' such that (1.) F' is a $\mathsf{MON}_=$ formula, (2.) $F' \equiv \exists p\, F$, (3.) p does not occur in F', (4.) All free variables, constants and predicates in F' do occur in F.*

The condition that all predicates in F' occur there only in polarities in which they also occur in F could also be added. The proposition implies that second-order quantifier elimination can be successfully performed for $\mathsf{QMON}_=$ with the following procedure: Replace subformulas of the form $\forall p\, G$ with $\neg\exists p\, \neg G$ and exhaustively rewrite subformulas of the form $\exists p\, G$ where G is a $\mathsf{MON}_=$ formula (i.e., $\exists p\, G$ is an innermost second-order quantification) to $\mathsf{MON}_=$ formulas according to Prop. 1. Satisfiability of a $\mathsf{QMON}_=$ formula F can be decided by applying this elimination method to

$$\exists p_1 \ldots \exists p_n\, \exists x_1 \ldots \exists x_m\, \exists c_1 \ldots \exists c_k\, F, \tag{i}$$

where $p_1, \ldots, p_n$ are all predicates with free occurrences in F, $x_1, \ldots, x_m$ are the free variables in F and $c_1, \ldots, c_k$ are the constants in F. The result is a $\mathsf{MON}_=$

sentence without any predicates and constants but possibly with equality. It can be transformed to a Boolean combination of basic formulas of the form $\exists^{\geq n}x\top$, which are satisfied by exactly those interpretations whose domain has at least n members. A Boolean combination of such basic formulas is then either true for all domain cardinalities with exception of a finite number or false for all domain cardinalities with exception of a finite number. The respective cardinalities can be read off easily from a representation in disjunctive normal form with $\exists^{\geq n}x\top$ in the role of atoms: each satisfiable conjunction then justifies a series of numbers with a lower limit or with lower as well as upper limits as domain cardinalities. For sufficiently large finite and for all infinite domains the value of the sentence is the same.

We now turn to the proof of Prop. 1 that is, to Behmann's method for second-order quantifier elimination by equivalence preserving formula rewriting. We make here only the characteristic steps of the method precise (see [49] for a more detailed account). For conversions that can be easily performed by rewriting with well-known equivalences only the effect is indicated. Some of the equivalences that are familiar from conversion to prenex form are now applied in the reverse direction, since in Behmann's method quantifiers are moved *inward* as far as possible, until theirs scopes do no longer overlap. A less common equivalence that is often applied is:

$$p(t) \;\equiv\; \forall x\, p(x) \lor x \neq t, \tag{ii}$$

for all constants or variables t different from x; dually $p(t) \equiv \exists x\, p(x) \land x = t$. The actual elimination steps are justified by the following equivalence:

Proposition 2 (Basic Elimination Lemma). *Let p be a unary predicate and let F, G be first-order formulas with equality in which p does not occur. Then*

$$\exists p\, (\forall x\, F \lor p(x)) \land (\forall x\, G \lor \neg p(x)) \;\equiv\; \forall x\, F \lor G.$$

Formulas F and G in that proposition may contain free occurrences of x, which are bound by the surrounding $\forall x$ on both sides. The goal of the elimination method is now to rewrite an input formula $\exists p\, F$, where F is a $\mathsf{MON}_{=}$ formula, such that all occurrences of quantification upon p match the left side of Prop. 2.

This is achieved by a conversion such that all subformulas starting with $\exists p$ are in a normalized form, called here *Generalized Eliminationshauptform* (Behmann calls a simpler variant for inputs without equality *Eliminationshauptform* [*main form for elimination*]). The following proposition shows this form and the conversion from it to applicability of Prop. 2. The counting quantifier $\forall^{<n}x$ is used there as shorthand for $\neg\exists^{\geq n}x\neg$:

Proposition 3 (From Generalized Eliminationshauptform to the Basic Elimination Lemma). *Let p be a unary predicate and let F be the formula*

$$\exists p \bigwedge_{1\leq i\leq a} (\forall x^{<a_i}\, A_i[x] \lor p(x)) \land \bigwedge_{1\leq i\leq b} (\forall x^{<b_i}\, B_i[x] \lor \neg p(x)) \land \\ \bigwedge_{1\leq i\leq c} (\exists x^{\geq c_i}\, C_i[x] \land p(x)) \land \bigwedge_{1\leq i\leq d} (\exists x^{\geq d_i}\, D_i[x] \land \neg p(x)),$$

where a, b, c, d are natural numbers ≥ 0, for the referenced values of i the a_i, b_i, c_i, d_i are natural numbers ≥ 1, and the $A_i[x], B_i[x], C_i[x], D_i[x]$ are first-order *formulas in which p does not occur. Then F is equivalent to*

$$Q\, G \land \exists p\, (\forall x\, A[x] \lor p(x)) \land (\forall x\, B[x] \lor \neg p(x)),$$

where Q is an existential quantifier prefix upon the following fresh variables: $x_{i1} \ldots x_{i(a_i-1)}$, $1 \leq i \leq a$; $y_{i1} \ldots y_{i(b_i-1)}$, $1 \leq i \leq b$; $u_{i1} \ldots u_{ic_i}$, $1 \leq i \leq c$; $v_{i1} \ldots v_{id_i}$, $1 \leq i \leq d$, where $G = \bigwedge_{1 \leq i \leq c,\, 1 \leq j \leq c_i} (C_i[u_{ij}] \land \bigwedge_{j < k \leq c_i} u_{ij} \neq u_{ik}) \land \bigwedge_{1 \leq i \leq d,\, 1 \leq j \leq d_i} (D_i[v_{ij}] \land \bigwedge_{j < k \leq d_i} v_{ij}, \neq v_{ik})$, with $C_i[u_{ij}]$ and $D_i[v_{ij}]$ denoting $C_i[x]$ and $D_i[x]$ after replacing all free occurrences of x by u_{ij} and v_{ij}, respectively, and where

$$A[x] = \bigwedge_{1 \leq i \leq a} (A_i[x] \lor \bigvee_{1 \leq j < a_i} x = x_{ij}) \land \bigwedge_{1 \leq i \leq c,\, 1 \leq j \leq c_i} x \neq u_{ij}, \text{ and}$$
$$B[x] = \bigwedge_{1 \leq i \leq b} (B_i[x] \lor \bigvee_{1 \leq j < b_i} x = y_{ij}) \land \bigwedge_{1 \leq i \leq d,\, 1 \leq j \leq d_i} x \neq v_{ij}.$$

The proof of Prop. 3 makes use of the different ways to expand counting quantifiers shown at the end of Sect. 1, such that for universal as well as existential counting quantifiers existential variables are produced which can be moved in front of the existential predicate quantifier. For example, $\forall x^{\overline{<a_i}}\, A_i[x] \lor p(x) \equiv \exists x_{i1} \ldots \exists x_{i(a_i-1)} \forall x\, (A_i[x] \lor \bigvee_{1 \leq j < a_i} x = x_{ij}) \lor p(x)$ and $\exists^{\geq c} x\, C_i[x] \land p(x) \equiv \exists u_{i1} \ldots \exists u_{ic} \bigwedge_{1 \leq j \leq c} (C_i[u_{ij}] \land \bigwedge_{j < k \leq c} u_{ij} \neq u_{ik}) \;\land\; \bigwedge_{1 \leq j \leq c} (\forall x\, x \neq u_{ij} \lor p(x))$. For inputs without equality, the *Eliminationshauptform* is sufficient:

$$\begin{aligned} \exists p\, &\textstyle\bigwedge_{1 \leq i \leq a} (\forall x\, A_i[x] \lor p(x)) \land \bigwedge_{1 \leq i \leq b} (\forall x\, B_i[x] \lor \neg p(x)) \;\land \\ &\textstyle\bigwedge_{1 \leq i \leq c} (\exists x\, C_i[x] \land p(x)) \land \bigwedge_{1 \leq i \leq d} (\exists x\, D_i[x] \land \neg p(x)), \end{aligned} \tag{iii}$$

It is equivalent to

$$\begin{aligned} \exists u_1 \ldots \exists u_c \exists v_1 \ldots \exists v_d\, &\textstyle\bigwedge_{1 \leq i \leq c} C_i[u_i] \land \bigwedge_{1 \leq i \leq d} D_i[v_i] \;\land \\ &\exists p\, \forall x\, ((\textstyle\bigwedge_{1 \leq i \leq a} A_i[x] \land \bigwedge_{1 \leq i \leq c} x \neq u_i) \lor p(x)) \;\land \\ &\forall x\, ((\textstyle\bigwedge_{1 \leq i \leq b} B_i[x] \land \bigwedge_{1 \leq i \leq d} x \neq v_i) \lor \neg p(x)), \end{aligned} \tag{iv}$$

where $u_1, \ldots, u_c$ and $v_1, \ldots, v_d$ are fresh variables. The result of eliminating p according to Prop. 2 then can be further rewritten to:

$$\begin{aligned} &(\forall x\, \textstyle\bigwedge_{1 \leq i \leq a} A_i[x] \;\lor\; \bigwedge_{1 \leq i \leq b} B_i[x]) \;\land \\ &\exists u_1 \ldots \exists u_c \exists v_1 \ldots \exists v_d\, \textstyle\bigwedge_{1 \leq i \leq c,\, 1 \leq j \leq d} u_i \neq v_j \;\land \\ &\quad \textstyle\bigwedge_{1 \leq i \leq c} (C_i[u_i] \land \bigwedge_{1 \leq j \leq b} B_j[u_i]) \land \bigwedge_{1 \leq i \leq d} (D_i[v_i] \land \bigwedge_{1 \leq j \leq a} A_j[v_i]), \end{aligned} \tag{v}$$

where $A_i[t]$, $B_i[t]$, $C_i[t]$, $D_i[t]$ denote $A_i[x]$, $B_i[x]$, $C_i[x]$, $D_i[x]$, respectively, with all free occurrences of x replaced by t. Equality enters in preparation of the form (iii) by rewriting occurrences of p with constant argument by (ii) and through handling existential quantifiers in proceeding from (iii) to (iv). The introduced equality literals actually either have a constant or two existential variables as arguments, implying that the simpler variant without dedicated equality handling is sufficient for elimination in formulas $\exists p_1 \ldots \exists p_n\, F$ where F is a MON formula (Behmann shows a special translation which is exponential in n for this case).

The conversion of $\exists p\, F$ to a form where all subformulas starting with $\exists p$ match the Generalized Eliminationshauptform of Prop. 3 proceeds in two steps. First the $\mathsf{MON}_=$ formula F is converted to a form where the quantifiers of instance variables are propagated inward such that their scopes do not overlap. We call such forms here *innex* as suggested by Behmann.[1] Achieving this form requires potentially *expensive* rewritings, in particular the distribution of conjunction over disjunction and vice versa, if this can effect further narrowing of quantifier scopes. Consider for example: $\forall x\, p(x) \lor (q(x) \land \exists y\, r(y)) \equiv \forall x\, (p(x) \lor q(x)) \land (p(x) \lor \exists y\, r(y)) \equiv (\forall x\, p(x) \lor q(x)) \land ((\forall x\, p(x)) \lor (\exists y\, r(y)))$. In automated reasoning, forms where quantifiers are propagated inward have also been considered, e.g. [16,37], but typically just as preprocessing operations, which would preclude the required expensive operations. In a variant of Behmann's method by Quine [38], the innex form is achieved by exhaustively rewriting innermost formulas with the following equivalence, shown here in dual variants:

$$\exists x F[G] \equiv (G \lor \exists x\, F[\bot]) \land (\neg G \lor \exists x\, F[\top]), \tag{vi}$$

$$\forall x F[G] \equiv (G \land \forall x\, F[\top]) \lor (\neg G \land \forall x\, F[\bot]), \tag{vii}$$

where $F[G]$ is a first-order formula with occurrences of a subformula G in which x does not occur free and whose free variables are not in scope of a quantifier within $F[G]$. Formulas $F[\top]$ and $F[\bot]$ denote $F[G]$ with all the occurrences of G replaced by $\top$ or $\bot$, respectively. Variant (vii) is a generalization of the well-known propositional Shannon expansion.

In presence of equality, the conversion to innex form introduces counting quantifiers by rewriting formulas of the form (viii) below to either (ix) or (x): Let $F[x]$ be a first-order formula in which variable x possibly occurs free, let $T = \{t_1, \ldots, t_n\}$ be an ordered set of n distinct constants or variables which are different from x and which do not occur in $F[x]$. Let $F[t]$ denote $F[x]$ with all free occurrences of x replaced by t. Then:

$$\exists x\, F[x] \land \textstyle\bigwedge_{1 \le i \le n} x \neq t_i \tag{viii}$$

$$\equiv \textstyle\bigvee_{1 \le m \le n} ((\exists^{\ge m} x\, F[x]) \land \mathsf{AUX}(m)) \;\lor\; \exists^{\ge n+1} x\, F[x] \tag{ix}$$

$$\equiv (\exists^{\ge 1} x\, F[x]) \;\land \textstyle\bigwedge_{1 \le m \le n} ((\exists^{\ge m+1} x\, F[x]) \lor \mathsf{AUX}(m)), \tag{x}$$

where $\mathsf{AUX}(m)$ stands for $\bigwedge_{S \subseteq T, |S| = m} (\bigvee_{t \in S} \neg F[t] \lor \bigvee_{t_i, t_j \in S, i<j} t_i = t_j)$. For example: $\exists x\, p(x) \land x \neq a \land x \neq b \;\equiv\; ((\exists^{\ge 1} x\, p(x)) \land \neg p(a) \land \neg p(b)) \lor ((\exists^{\ge 2} x\, p(x)) \land (\neg p(a) \lor \neg p(b) \lor a = b)) \lor \exists^{\ge 3} x\, p(x) \;\equiv\; \exists^{\ge 1} x\, p(x) \land ((\exists^{\ge 2} x\, p(x)) \lor (\neg p(a) \land \neg p(b))) \;\land ((\exists^{\ge 3} x\, p(x)) \lor \neg p(a) \lor \neg p(b) \lor a = b)$.

The result of the innex conversion with respect to quantifiers upon instance variables is captured in the following proposition:

Proposition 4 (Counting Quantifier Innex Form for $\mathsf{MON}_=$ Formulas). *There is an effective method to compute from a given $\mathsf{MON}_=$ formula F a formula F' such that: (1.) F' is a Boolean combination of basic formulas of the form: (a) p, where p is a nullary predicate, (b) $p(t)$, where p is a unary predicate and t is a constant or an variable, (c) $t = s$, where each of t, s is a constant or a variable, (d) $\exists^{\ge n} x \bigwedge_{1 \le i \le m} L_i[x]$, where $n \ge 1$, $m \ge 0$ and the $L_i[x]$ are pairwise different and pairwise non-complementary positive or negative literals with a unary predicate applied to the variable x. (2.) $F' \equiv F$. (3.) All free variables, constants and predicates in F' do occur in F.*

[1] Letter to Church, 30 Jan 1959 [6, Kasten 1, I 11].

If the given formula F is without equality, the allowed basic formulas can be strengthened by excluding the case $t = s$ (c) and restricting the case (d) to $n = 1$, such that the counting quantifier can be considered as standard quantifier.

The second step in converting $\exists p\, F$ leads from $\exists p\, F'$, where F' is a Boolean combination according to Prop. 4 to a formula where all subformulas starting with $\exists p$ match the Generalized Eliminationshauptform of Prop. 3. This can be achieved by first moving negation in F' inward followed by replacing formulas of the form $\neg\exists^{\geq n} x \bigwedge_{1\leq i\leq m} L_i[x]$ with $\forall^{<n} x \bigvee_{1\leq i\leq m} \overline{L_i[x]}$, where $\overline{L}$ denotes the complement of literal L. Then $\exists p$ is propagated inward with the same technique that had been applied to first-order quantifiers: $\exists p$ is distributed over disjunction, conjunctions are reordered such that conjuncts without p can be moved out of its scope, and – the potentially expensive – distribution of conjunction over disjunction is applied if that enables further distribution of $\exists p$ over disjunction.

3 Useful Second-Order Properties

The use of transformations that introduce auxiliary definitions, like the Tseitin and Plaisted-Greenbaum encoding, is common practice to obtain small equi-satisfiable conjunctive normal forms. Second-order quantification allows to understand the introduction and elimination of such definitions as equivalence preserving operations, with Ackermann's Lemma as a special case. The more fine grained account of semantics (instead of just equi-satisfiability) justifies the application of these techniques in elimination tasks. We compile these principles here for the case where the defined/eliminated predicates are unary.

Unless specially noted, we consider here formulas of first-order logic with equality. If p does not occur in F, then by Prop. 2 it holds that $\exists p \forall x\, p(x) \leftrightarrow F \equiv \top$. This allows to derive the following proposition:

Proposition 5 (Introduction and Elimination of Definitions). *Let p be a unary predicate, let x be an variable and let $G[x]$ be a formula in which p does not occur. For a constant or variable t, let $G[t]$ denote $G[x]$ with all free occurrences of x replaced by t. Let $F[G[t_1], \ldots, G[t_n]]$ be a formula in which p does not occur and which has n occurrences of subformulas, instantiated with $G[t_1], \ldots, G[t_n]$, respectively, neither of them in a context where a variable that occurs free in $G[x]$ is bound. Let $F[p(t_1), \ldots, p(t_n)]$ denote the same formula with the indicated occurrences $G[t_i]$ replaced by $p(t_i)$. Then*

$$F[G[t_1], \ldots, G[t_n]] \;\equiv\; \exists p\, (\forall x\, p(x) \leftrightarrow G[x]) \wedge F[p(t_1), \ldots, p(t_n)].$$

Prop. 5 can be applied from left to right to introduce auxiliary predicates p and from right to left to expand them, by replacing *all* occurrences of p with their definientia and then dropping the definition. If p occurs in $F[p(t_1), \ldots, p(t_n)]$ just with, say, positive polarity, then $\exists p\, (\forall x\, p(x) \leftrightarrow G[x]) \wedge F[p(t_1), \ldots, p(t_n)] \equiv \exists p\, (\forall x\, p(x) \rightarrow G[x]) \wedge F[p(t_1), \ldots, p(t_n)]$. This leads to Ackermann's Lemma [1]:

Proposition 6 (Ackermann's Lemma). *Assume the setting of Prop. 5 and that all the indicated subformula occurrences in $F[G[t_1], \ldots, G[t_n]]$ (or, equivalently, in $F[p(t_1), \ldots, p(t_n)]$) have the same polarity P. Then*

$$\exists p\, (\forall x\, p(x) \rightarrow G[x]) \wedge F[p(t_1), \ldots, p(t_n)] \equiv F[G[t_1], \ldots, G[t_n]], \;\; \textit{if } P \textit{ is positive.}$$
$$\exists p\, (\forall x\, p(x) \leftarrow G[x]) \wedge F[p(t_1), \ldots, p(t_n)] \equiv F[G[t_1], \ldots, G[t_n]], \;\; \textit{if } P \textit{ is negative.}$$

The Basic Elimination Lemma Prop. 2 is obviously an instance of Ackermann's Lemma. Vice versa, Ackermann's Lemma can be proven such that the only elimination step is performed according to Prop. 2.

In [2], a short sequel to [1], Ackermann shows a precondition which allows to move existential predicate quantification to the right of universal individual quantification, where the arity of the quantified predicate is reduced:

Proposition 7 (Ackermann's Quantifier Switching). *Let p be a predicate with arity $n{+}1$, where $n \geq 0$. Let $F = F[p(x, t_{11}, \ldots, t_{1n}), \ldots, p(x, t_{m1}, \ldots, t_{mn})]$, where $m \geq 1$, be a formula of second-order logic in which p has the exactly m indicated occurrences. Assume further that p and x occur only free in F. Let q be a predicate with arity n that does not occur in F and let $F[q(t_{11}, \ldots, t_{1n}), \ldots, q(t_{m1}, \ldots, t_{mn})]$ denote F with each occurrence $p(x, t_{ij}, \ldots, t_{ij})$ of p replaced by $q(t_{ij}, \ldots, t_{ij})$, for $1 \leq i \leq n$, $1 \leq j \leq m$. Under the assumption of the axiom of choice it then holds that*

$$\begin{aligned} &\exists p \forall x\, F[p(x, t_{11}, \ldots, t_{1n}), \ldots, p(x, t_{m1}, \ldots, t_{mn})] \\ \equiv\ &\forall x \exists q\, F[q(t_{11}, \ldots, t_{1n}), \ldots, q(t_{m1}, \ldots, t_{mn})]. \end{aligned}$$

Ackermann applies this equivalence in [2] to avoid Skolemization and to convert formulas such that monadic techniques or Ackermann's Lemma become applicable. Van Benthem [7, p. 211] mentions this equivalence with application from right to left to achieve prenex form w.r.t. second-order quantifiers.

4 Hidden Monadicity in Description Logics

The second-order properties compiled in Sect. 3 give us a toolkit to convert a *knowledge base* (KB), i.e., a TBox combined with an ABox, in the expressive description logic (DL) $\mathcal{ALCOQH}$ ($\mathcal{ALC}$ with nominals, qualified number restrictions and subroles) to an equi-satisfiable $\mathsf{QMON}_=$ formula. Given the decidability of $\mathsf{QMON}_=$ formulas, this provides a very simple proof of the decidability of the description logic. It also follows that any method to decide $\mathsf{QMON}_=$ formulas provides a decision method for the DL.

It is well-know that for many DLs, including $\mathcal{ALCOQH}$, a KB can be straightforwardly translated into a first-order formula (e.g., [40,23]) based on the standard translation of modal logics (e.g., [8]). We call this representation of a DL KB its standard first-order translation. It captures not just satisfiability but the full semantics of the KB. The standard first-order translation can be converted to a generalized conjunctive normal form, where the role of literals is played by basic formulas of certain forms. A structural normal form conversion, which involves introduction of auxiliary predicate definitions according to Prop. 5 can prevent the blow-up through distribution of disjunction over conjunction, can ensure that variables are introduced only in a limited way and can effect further normalization. If the translation proceeds by expanding equivalences corresponding to definitional TBox axioms into implications and conversion to negation normal form, Ackermann's Lemma (6) is sufficient to justify the introduction of the auxiliary predicates, corresponding to the Plaisted-Greenbaum encoding. (See [23] for a thorough presentation of such structure preserving translations of description logics into specific decidable first-order fragments.) For the standard

Table 1. Forms of basic formulas in DL normalizations. The symbols c, d and r match unary or binary predicates, respectively. Variables are understood literally as shown.

	Form	*Inducing DL construct*
1.	$c(x)$	atomic concept, ABox assertion
2.	$\neg c(x)$	atomic concept
3.	$\exists y\, r(x,y) \land d(y)$	qualified existential restriction
4.	$\forall y\, \neg r(x,y) \lor d(y)$	qualified value restriction
5.	$x = a$	nominal
6.	$x \neq a$	nominal, ABox assertion
7.	$r(x,a)$	ABox assertion
8.	$\forall y\, \neg r(x,y) \lor r(x,y)$	subrole
9.	$\exists^{\geq n} y\, r(x,y) \land d(y)$	qualified number restriction
10.	$\neg(\exists^{\geq n} y\, r(x,y) \land \neg d(y))$	qualified number restriction

first-order translation of an $\mathcal{ALCOQH}$ KB this normalization yields an equivalent second-order formula $\exists d_1 \ldots \exists d_k \forall x\, F$, where $d_1, \ldots, d_k$ are fresh unary auxiliary predicates and F is a first-order conjunction of disjunctions of basic formulas of the forms shown in Table 1.

In the conversion of ABox assertions equivalence (ii) is involved. The counting quantifiers can be considered as abbreviations for formulas as shown at the end of Sect. 1. The translation $\exists d_1 \ldots \exists d_k \forall x\, F$ is equi-satisfiable with the following second-order formula:

$$\exists c_1 \ldots \exists c_n\, \exists r_1 \ldots \exists r_m\, \exists d_1 \ldots \exists d_k\, \forall x\, F, \tag{xi}$$

where $c_1, \ldots, c_n$ are the unary predicates in F with exception of the $d_1, \ldots, d_k$ (corresponding to names of atomic concepts in the KB) and $r_1, \ldots, r_m$ are the binary predicates in F (corresponding to role names in the KB). The predicate quantifiers can be reordered such that $\exists r_1 \ldots \exists r_m$ immediately precedes $\forall x$. Since all occurrences of $r_1, \ldots, r_m$ in F have x as first argument, by Prop. 7 formula (xi) is equivalent to

$$\exists c_1 \ldots \exists c_n\, \exists d_1 \ldots \exists d_k\, \forall x\, \exists r'_1 \ldots \exists r'_m\, F', \tag{xii}$$

where the $r'_1, \ldots, r'_m$ are fresh unary predicates and F' is obtained from F by replacing for all $i \in \{1, \ldots, m\}$ all occurrences of the form $r_i(x,t)$, where t is some term, with $r'_i(t)$.

Formula (xii) is a $\mathsf{QMON}_=$ formula. If no number restrictions are involved, the effort required by this translation is linear in the size of the original KB. Otherwise, the expansion of the counting quantifiers into first-order logic has to be taken into account, whose size is linear in the cardinality argument of the quantifier. The following theorem statement summarizes what has been shown:

Theorem 8 (Reduction of $\mathcal{ALCOQH}$ Knowledge Base Satisfiability to Satisfiability of Relational Monadic Formulas). *Under assumption of the axiom of choice, there is a polynomial time translation from an $\mathcal{ALCOQH}$ knowledge base to an equi-satisfiable* $\mathsf{QMON}_=$ *sentence. The translation takes time linear in the size of the standard first-order translation of the knowledge base.*

An elimination-based decision procedure may yield requirements on the domain cardinality. This applies also to translated DL KBs. A simple example is the KB $\{\top \sqsubseteq \exists r.c,\ \top \sqsubseteq \exists r.\neg c\}$. We obtain that the KB is only satisfiable for domains whose cardinality is at least two: $\exists c \forall x \exists r'\, (\exists y\, r'(y) \land c(y)) \land (\exists y\, r'(y) \land \neg c(y)) \equiv \exists y \exists z\, y \neq z \equiv \exists^{\geq 2} x \top$. The KB $\{\top \sqsubseteq \{a\}\}$ translates into the equi-satisfiable $\exists a \forall x\, a = y$ (without predicate to eliminate), which can be expressed as $\neg \exists^{\geq 2} x \top$.

The $\mathsf{QMON}_=$ translation in formula (xii) suggests that the decision method has to proceed by first eliminating the role predicates $r'_1, \ldots, r'_m$, before any of the concept predicates can be eliminated. One further conversion step can be applied to relax this by also moving those of the other quantified predicates that only occur with x as argument in F' to the right of $\forall x$ with Prop. 7. The introduction of auxiliary predicates in the processing of the standard first-order translation can be arranged such that this applies to all predicates that correspond to concept names in the input KB (the initial normalization of the resolution-based elimination method in [27] satisfies an analogous criterion). The resulting translation is then a $\mathsf{QMON}_=$ formula of the form

$$\exists d_1 \ldots \exists d_k\, \forall x\, \exists r'_1 \ldots \exists r'_m\, \exists c'_1 \ldots \exists c'_n\, F'', \tag{xiii}$$

where the $c'_1, \ldots, c'_n$ are fresh nullary predicates, and F'' is obtained from F' in (xii) by replacing for all $i \in \{1, \ldots, n\}$ all occurrences of $c_i(x)$ with c'_i.

As we have seen, elimination of *all* concept and role predicates can be successively performed to decide $\mathcal{ALCOQH}$ knowledge bases. We now consider actual elimination problems, where just *some* predicates should be eliminated. Given is the standard first-order translation K of a knowledge base and a set $\{p_1, \ldots, p_n\}$ of unary predicates that represent concept names in the knowledge base. The objective is to apply second-order quantifier elimination to

$$\exists p_1 \ldots p_n\, K. \tag{xiv}$$

The normalization with auxiliary predicates described above for deciding satisfiability and further straightforward equivalence preserving conversion then yield a formula that is equivalent to (xiv) and has the following form:

$$S \land \exists c_1 \ldots \exists c_l\, \exists d_1, \ldots, \exists d_k\, \forall x\, F, \tag{xv}$$

where the $c_1, \ldots, c_l$ are those unary predicates in F that only occur with x as argument which includes the $p_1, \ldots, p_n$, the $d_1, \ldots, d_k$ are all the remaining unary auxiliary predicates introduced in the normalization, S is a sentence in which the binary predicates $r_1, \ldots, r_m$ representing roles in F are the only predicates, and F is a conjunction of disjunctions of basic formulas as displayed in Table 1.

The S component can in particular be used to express inverse roles by formulas like $\forall x\, \forall y, r_i(x, y) \leftrightarrow r_j(y, x)$. Let $R[x]$ be the formula $\bigwedge_{1 \leq i \leq m} (\forall y\, r'_i(y) \leftrightarrow r_i(x, y))$. By Prop. 5, formula (xv) is then equivalent to

$$S \land \exists c_1 \ldots \exists c_l\, \exists d_1, \ldots, \exists d_k\, \forall x\, \exists r'_1 \ldots \exists r'_m\, R[x] \land F', \tag{xvi}$$

where, as in formula (xii), the $r'_1, \ldots, r'_m$ are fresh unary predicates and F' is obtained from F by replacing all occurrences of $r_i(x, t)$ with $r'_i(t)$. By arguments analogously to the derivation of formula (xiii), formula (xvi) is equivalent to:

$$S \land \exists d_1, \ldots, \exists d_k\, \forall x\, \exists r'_1 \ldots \exists r'_m\, R[x] \land \exists c'_1 \ldots \exists c'_l\, F'', \tag{xvii}$$

where, as in formula (xvi), the $c'_1, \ldots, c'_l$ are fresh nullary predicates and F'' is obtained from F' by replacing all occurrences of $c_i(x)$ with c'_i. Clearly, F'' is a $\mathsf{MON}_=$ formula, implying that $\exists c'_1 \ldots \exists c'_l$ can be successfully eliminated by monadic techniques. The $\exists r'_1 \ldots \exists r'_m$ can then be linearly eliminated according to Prop. 5 by unfolding their definitions in $R[x]$, followed by removing $R[x]$.

If $k = 0$, that is, there are no $\exists d_i$, which is evidently the case if among the constructs in Table 1 only the limited versions of restriction are permitted, that is, in lines 3. and 9. of the table only $\top$ is allowed in place of $d(y)$ and in line 4. and 10. only $\bot$, then the elimination is now completed. This result is expressed in the following theorem statement.

Theorem 9 (Monadic Concept Elimination in DLs with Limited Restriction). *We consider knowledge bases expressed in a description logic that is like $\mathcal{ALCOQH}$ but only allows limited restriction and allows in addition inverse roles. Under the assumption of the axiom of choice, there is a linear time translation that converts the standard first-order translation K of such a knowledge base and a set $\{p_1, \ldots, p_n\}$ of unary predicates representing concept names in K to a relational second-order formula that is equivalent to $\exists p_1 \ldots \exists p_n\, K$ and such that those second-order quantifiers whose argument is not a $\mathsf{QMON}_=$ formula can be eliminated linearly by a series of applications of Prop. 5.*

With permitting inverse roles but only limited restriction, the description logics covered by Theorem 9 include the typical representatives of the DL-Lite family [10]. The theorem can be easily strengthened to allow also the forgetting of roles whose inverse is not used (more generally: whose corresponding predicates r_i do not occur in the S component of (xv)). To achieve this, the definitions of their corresponding unary predicates r'_i have just to be omitted from $R[x]$.

An obvious limit of the translation underlying Theorem 9 is that elimination of the $\exists d_1 \ldots \exists d_k$ in (xvii) with techniques based on monadicity is blocked: the argument formula of the $\exists d_i$ contains with $R[x]$ binary predicates, and Prop. 7 can not be applied to move the d_i to the right of $\forall x$ (and of $R[x]$) because they occur in F'' with arguments other than x. So far, a general technique to overcome this in the monadic setting has not been developed. In particular situations, elimination of an $\exists d_i$ might nevertheless be possible after eliminating the $\exists c'_1 \ldots \exists c'_k$. For example, if all occurrences of d_i then have x as argument, possibly also after switching names of universal variables in some conjuncts, or after introducing additional fresh d_i predicates (which may lead to non-termination). Also the inclusion of other elimination techniques seems possible, in particular of ones that can be considered as simplifications such as elimination in the case where d_i occurs just in a single polarity. A further option might be to accept predicates d_i in the elimination output if they can be regarded as just encoding formula structure.

With the approach of elimination in description logics via embedding into first-order logic, the issue of re-translation of the first-order elimination result to the source language arises. Further auxiliary unary predicates introduced according to Prop. 5 might be helpful to encapsulate complex basic formulas that should not be broken during elimination. A general question is, how to deal with source languages whose first-order consequences diverge from the consequences expressible in the language. As we have already seen, eliminating r and c from $\{\top \sqsubseteq \exists r.c,\ \top \sqsubseteq \exists r.\neg c\}$ yields the first-order consequence $\exists^{\geq 2} x \top$, which as such can not be expressed by an $\mathcal{ALC}$ KB. If the elimination result should be combined with another knowledge base, say, $\{\top \sqsubseteq \{a\}\}$, it does well matter whether the consequence $\exists^{\geq 2} x \top$ is retained. Related examples, where forgetting in $\mathcal{ALC}$ ontologies yields results with number restrictions that are expressed as $\mathcal{SHQ}$ ontologies, can be found in [28].

5 Direct Methods in View of Monadic Techniques

Direct methods (also called methods following the *Ackermann approach*) were introduced with the DLS algorithm [15,22,11] that operates on the basis of first-order formulas. Its preprocessing step tries to rewrite the input such that all innermost occurrences of second-order quantifiers allow elimination by Ackermann's Lemma. A comparison of DLS with Behmann's innex conversion immediately suggests an improvement of DLS: The preprocessing of DLS starts with conversion to negation normal form and does not include a rule to distribute disjunction over conjunction. (It does includes a rule to distribute conjunction over disjunction.) A simple example where DLS fails unnecessarily because no preprocessing rule is applicable is thus $\exists p \forall x\, (p(x) \land q(x)) \lor (\neg p(x) \land r(x))$.

It thus seems that DLS should be enhanced with distributing disjunction over conjunction or equivalent techniques. In contrast to the original [15] and the carefully analyzed variant [11] of DLS, related enhancements have been considered for the implementation [22], but not in a systematic way. A recent direct method for modal logics [41] has a single rule which covers both required forms of distribution since it does not operate on negation normal form.

Algorithms based on Ackermann's Lemma operate by preprocessing the input such that all innermost occurrences of second-order quantifiers are in formulas of the form $\exists p\, F_1 \land F_2$, where p occurs in F_1 only in positive and in F_2 only in negative polarity. This form can always be converted into two alternate forms where each subformula that starts with $\exists p$ matches the left side of the first or second variant of Ackermann's Lemma, respectively. However, this step might involve the introduction of Skolem functions that have to be replaced after eliminating p by existential variables, which is not possible in all cases. If one of the conjuncts F_1 or F_2 is a $\mathsf{MON}_=$ formula, then this rewriting can be performed without introduction of Skolem functions guaranteeing successful elimination with Ackermann's Lemma because there is no need for potentially failing un-Skolemization. Based on the techniques from Sect. 2, the conversion of $\exists p\, F_1 \land F_2$ can be achieved as follows for the case where F_1 is a $\mathsf{MON}_=$ formula: Let k be a fresh nullary predicate. Convert $\exists p\ F_1 \land k$ to Behmann's Generalized Eliminationshauptform (Prop. 3) without applying rewritings which depend on the fact that p does not occur in k. Replace all occurrences of k with F_2. This shows the following statement:

Theorem 10 (Applicability of Ackermann's Lemma on Semi Monadic Formulas). *Consider a formula $\exists p\ F_1 \land F_2$ where F_1 is a $\mathsf{MON}_=$ formula in which p occurs only with positive polarity and F_2 is a first-order formula in* *which p only occurs with negative polarity. Then $\exists p\; F_1 \land F_2$ is equivalent to a second-order formula in which all occurrences of second order quantifiers are upon p and are of the form $\exists p\,(\forall x\, F_1' \rightarrow p(x)) \land F_2'$ where F_1' is a $\mathsf{MON}_=$ formula without any occurrence of p and F_2' a first-order formula with only negative occurrences of p. Moreover, all free variables, constants and predicates in formulas F_1' and F_2' occur already in $F_1 \land F_2$. This statement applies analogously for the case where p occurs with the respective complementary polarities in F_1 and F_2.*

Successful termination on all elimination tasks that express the computation of frame correspondence properties of Sahlqvist formulas is a desired and investigated property of elimination methods [21,11,13,41]. The Sahlqvist-van Benthem substitution algorithm [39,7,8] is a specialized method for that problem, where an involved substitution step can be considered as elimination with Ackermann's Lemma. Based on the presentation in of that algorithm in [8, Sect. 3.6], we show that the success of elimination for Sahlqvist formulas can be attributed in part to the fact that a match with the "semi monadic" case, the precondition of Theorem 10 can be established. We consider here just the core step of the Sahlqvist-van Benthem algorithm. Given is a formula of the form:

$$\forall p_1 \ldots \forall p_n\; \forall x_1 \ldots \forall x_m\, (\mathrm{REL} \land \text{BOX-AT}) \rightarrow \mathrm{POS}, \qquad \text{(xviii)}$$

where REL is a conjunction of atomic statements of the form $r(x_i, x_j)$, BOX-AT is a conjunction of formulas of the form $\forall y\, r_\beta(x_i, y) \rightarrow p(y)$ and POS is a formula in which $p_1 \ldots p_n$ only occur with positive polarity. The $r_\beta(x_i, y)$ abbreviate formulas which have x_i and y as only free variables and which do not contain any of the $p_1, \ldots, p_n$. (The r_β could be formally introduced according to Prop. 5). Formula xviii is equivalent to:

$$\neg\exists x_1 \ldots \exists x_m\, \mathrm{REL} \land \exists p_1 \ldots \exists p_n\, \text{BOX-AT} \land \neg\mathrm{POS}. \qquad \text{(xix)}$$

Conjunct $\neg$POS is an arbitrary formula in which the $p_1, \ldots, p_n$ occur only negatively. Conjunct BOX-AT is a conjunction of formulas $\forall y\, r_\beta(x_i, y) \rightarrow p(y)$. Thus, the $p_1, \ldots, p_n$ only occur only positively in BOX-AT. Prop. 5 allows to replace the r_β by unary predicates r'_{β_i} defined with formulas $\exists r'_{\beta_i}(\forall y\, r'_{\beta_i}(y) \leftrightarrow r_\beta(x_i, y))$ interspersed immediately before $\exists p_1 \ldots \exists p_n$. The preconditions of Theorem 10 are now met for the subformula starting at $\exists p_n$ and can be similarly established with respect to $\exists p_{n-1}$ after factoring implications $\forall y\, r_{\beta_i}(y) \rightarrow p_n(y)$ and eliminating $\exists p_n$ with Ackermann's Lemma. The auxiliary predicates r'_{β_i} can be expanded again by Prop. 5 after eliminating $\exists p_1 \ldots \exists p_n$.

6 Related Work

In [27,26,28,29] methods for uniform interpolation in various expressive DLs are presented, which are explicitly related to resolution based elimination and Ackermann's Lemma. They are based on a conjunctive normal form translation with auxiliary defined concepts analogous to that described in Sect. 4 and operate in two phases, related to the problem of eliminating the $\exists d_i$ exhibited in formula (xvii). In a resolution-based first phase at least all input concepts that should be forgotten are eliminated. In this phase a finite (but possibly exponential) number of fresh auxiliary concepts is introduced in a controlled way. This phase is sufficient to decide the formula. A normalization is preserved such that in the second phase all the remaining auxiliary concepts can be eliminated either by Ackermann's Lemma, or, in case of circular dependency, by a fixpoint generalization of it [36]. The preserved normalization ensures re-translatability of the results to fixpoint extensions of the respective DL. Monadic properties have not been explicitly considered in these works, but might be implicit in the used normal form which represents concept and role names by propositional symbols.

In [3] equi-satisfiable translations of variants of DL-Lite into the one-variable fragment of first-order logic are developed. Elimination problems have not been considered there. The translation is not systematically derived by using second-order equivalences. It needs to be investigated, whether its representation of inverse roles and number restrictions can be transferred to the setting of Sect. 4. Forgetting and related concepts are investigated for DL-Lite in [25], a specialized algorithm for concept forgetting in DL-Lite is shown in [45].

Alternative decision methods for MON formulas include resolution: Equipped with an appropriate ordering and condensation, it decides MON formulas, although the associated Herbrand universe might be infinite due to Skolemization [17]. A superposition-based decision method for $\mathsf{MON}_=$ is given in [4]. Deciding satisfiability for MON and $\mathsf{MON}_=$ is $\mathsf{NEXPTIME}$-complete, as presented in [9, Sect. 6.2] along with more fine-grained results. The method of [30] underlying the upper bound verifies a given interpretation by repeatedly constructing an innex form with respect to some innermost individual quantifier occurrence and then replacing the corresponding obtained quantified subformulas with $\top$ or $\bot$ according to the interpretation. Only atoms present in the input are involved.

Relational monadic formulas have applications in verification: In [42] a decision method for S1S, applied in the verification of temporal properties, is described, which involves conversion to Behmann's innex form. An OBDD-based implementation is mentioned there. In [44] techniques to detect whether polyadic relations correspond to a finite union of Cartesian products and, if this is the case, decompose them into monadic form are developed.

7 Conclusion

We have restored the historic method by Behmann for second-order quantifier elimination over a fragment of first-order logic, relational monadic formulas, where elimination succeeds in general. It has striking similarities with the direct approach of modern elimination methods, which are based on the more powerful Ackermann's Lemma that also applies to formulas with polyadic predicates and functions, but do not succeed in the general case. We moved on to inspect some applications of elimination with the conjecture that monadicity might play a role in their success, in particular with a quantifier switching technique devised by Ackermann to extend the applicability of methods for monadic formulas and of the lemma named after him. A review of description logics viewed as embedded into first-order logic shows that the decision problem for expressive logics such as $\mathcal{ALC}$ can be reduced to the decision problem for relational monadic formulas with second-order quantification. While the corresponding elimination of all role and concept symbols succeeds, the structure of the translation prevents the elimination of just an arbitrary selection of concept symbols. For elimination in description logics of the DL-Lite family this provides no obstacle.

The involved transformations are all obtained from the standard relational first-order translation with equivalence preserving steps that make use of a few specific second-order equivalences. This is a clear and safe methodology which suggests to investigate possibilities of mechanization, for example to detect cases of eliminability or decidability that are not apparent in the syntactic form.

A further observation was that on a formula that has been separated by a direct method in preparation for Ackermann's Lemma the elimination can be safely performed if one of the separated components is a monadic relational formula. The application to Sahlqvist formulas provides an instance of this case. It needs to be investigated whether the observation leads to completeness results for interesting classes that have not been considered previously.

Another issue for future research is the deeper investigation of methods. In particular the shown variant of quantifier innexing by Quine resembles methods of knowledge compilation based on the Shannon expansion [35,46]. For inputs from particular applications such as translated description logic knowledge bases it can be observed that they are already in innex from with respect to first-order quantifiers. A question that arises here is whether known special methods would be simulated by rewriting-based elimination methods.

Acknowledgements. This work was supported by DFG grant WE 5641/1-1.

References

1. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Math. Ann. 110, 390–413 (1935)
2. Ackermann, W.: Zum Eliminationsproblem der mathematischen Logik. Math. Ann. 111, 61–63 (1935)
3. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. JAIR (36), 1–69 (2009)
4. Bachmair, L., Ganzinger, H., Waldmann, U.: Superposition with simplification as a decision procedure for the monadic class with equality. In: KGC'93. LNCS, vol. 713, pp. 83–96 (1993)
5. Behmann, H.: Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem. Math. Ann. 86(3–4), 163–229 (1922)
6. Behmann, H. (Bestandsbildner): Nachlass Heinrich Johann Behmann, Staatsbibliothek zu Berlin – Preußischer Kulturbesitz, Handschriftenabt., Nachl. 335
7. van Benthem, J.: Modal Logic and Classical Logic. Bibliopolis (1983)
8. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambr. Univ. Press (2001)
9. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Springer (1997)
10. Calvanese, D., et al.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. JAR 39(3), 385–429 (2007)
11. Conradie, W.: On the strength and scope of DLS. JANCL 16(3–4), 279–296 (2006)
12. Conradie, W., Goranko, V., Vakarelov, D.: Algorithmic correspondence and completeness in modal logic. I. The core algorithm SQEMA. LMCS 2(1:5), 1–26 (2006)

13. Conradie, W., Goranko, V., Vakarelov, D.: Elementary canonical formulae: A survey on syntactic, algorithmic, and model-theoretic aspects. In: Advances in Modal Logic, vol. 5, pp. 17–51. College Pub. (2005)
14. Craig, W.: Elimination problems in logic: A brief history. Synthese (164), 321–332 (2008)
15. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited: A reduction algorithm. JAR 18(3), 297–338 (1997)
16. Egly, U.: On the value of antiprenexing. In: LPAR '94. LNCS, vol. 822, pp. 69–83 (1994)
17. Fermüller, C., Leitsch, A., Hustadt, U., Tammet, T.: Resolution decision procedures. In: Robinson, A., Voronkov, A. (eds.) Handb. of Autom. Reasoning, vol. 2, pp. 1793–1849. Elsevier (2001)
18. Gabbay, D.M., Schmidt, R.A., Szałas, A.: Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Pub. (2008)
19. Gabbay, D., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In: KR'92. pp. 425–435. Morgan Kaufmann (1992)
20. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: KR'06. pp. 187–197. AAAI Press (2006)
21. Goranko, V., Hustadt, U., Schmidt, R.A., Vakarelov, D.: SCAN is complete for all Sahlqvist formulae. In: RelMiCS 7. LNCS, vol. 3051, pp. 149–162 (2004)
22. Gustafsson, J.: An implementation and optimization of an algorithm for reducing formulae in second-order logic. Tech. Rep. LiTH-MAT-R-96-04, Univ. Linköping (1996)
23. Hustadt, U., Schmidt, R.A., Georgieva, L.: A survey of decidable first-order fragments and description logics. JoRMiCS 1, 251–276 (2004)
24. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: IJCAI09. pp. 830–835 (2009)
25. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. AI 174(15), 1093–1141 (2010)
26. Koopmann, P., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCH}$-ontologies. In: LPAR-19. LNCS (LNAI), vol. 8312, pp. 552–567. Springer (2013)
27. Koopmann, P., Schmidt, R.A.: Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In: FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 87–102. Springer (2013)
28. Koopmann, P., Schmidt, R.A.: Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In: IJCAR 2014. LNCS (LNAI), vol. 8562, pp. 434–448. Springer (2014)
29. Koopmann, P., Schmidt, R.A.: Uniform interpolation and forgetting for $\mathcal{ALC}$ ontologies with ABoxes. In: AAAI 15. pp. 175–181. AAAI Press (2015)
30. Lewis, H.R.: Complexity results for classes of quantificational formulas. Journal of Computer and System Sciences 21, 317–353 (1980)
31. Ludwig, M., Konev, B.: Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In: KR'14. AAAI Press (2014)
32. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: IJCAI-11. pp. 989–995. AAAI Press (2011)
33. Löwenheim, L.: Über Möglichkeiten im Relativkalkül. Math. Ann. 76, 447–470 (1915)
34. Mancosu, P., Zach, R.: Heinrich Behmann's 1921 lecture on the algebra of logic and the decision problem. The Bulletin of Symbolic Logic 21, 164–187 (2015)
35. Murray, N.V., Rosenthal, E.: Tableaux, path dissolution and decomposable negation normal form for knowledge compilation. In: TABLEAUX 2003. LNCS (LNAI), vol. 2796, pp. 165–180 (2003)

36. Nonnengart, A., Szałas, A.: A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In: Orlowska, E. (ed.) Logic at Work. Essays Ded. to the Mem. of Helena Rasiowa, pp. 89–108. Springer (1998)
37. Nonnengart, A., Weidenbach, C.: Computing small clause normal forms. In: Robinson, A., Voronkov, A. (eds.) Handb. of Autom. Reasoning, vol. 1, pp. 335–367. Elsevier (2001)
38. Quine, W.V.: On the logic of quantification. JSL 10(1), 1–12 (1945)
39. Sahlqvist, H.: Completeness and correspondence in the first and second order semantics for modal logic. In: Proc. Third Scand. Logic Symp. Uppsala (1973). pp. 110–143. North-Holland (1975)
40. Sattler, U., Calvanese, D., Molitor, R.: Relationships with other formalisms. In: Baader, F. et al. (ed.) The Description Logic Handb., pp. 137–177. Cambr. Univ. Press (2003)
41. Schmidt, R.A.: The Ackermann approach for modal logic, correspondence theory and second-order reduction. JAL 10(1), 52–74 (2012)
42. Schneider, K.: Verification of Reactive Systems. Springer (2003)
43. Skolem, T.: Untersuchungen über die Axiome des Klassenkalküls und über Produktations- und Summationsprobleme welche gewisse Klassen von Aussagen betreffen. Videnskapsselskapets Skrifter I. Mat.-Nat. Klasse(3) (1919)
44. Veanes, M., Bjørner, N., Nachmanson, L., Bereg, S.: Monadic decomposition. In: CAV 2014. LNCS, vol. 8559, pp. 628–645 (2014)
45. Wang, Z., Wang, K., Topor, R.W., Pan, J.Z.: Forgetting for knowledge bases in DL-Lite. Ann. Math. Artif. Intell 58, 117–151 (2010)
46. Wernhard, C.: Tableaux for projection computation and knowledge compilation. In: TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 325–340 (2009)
47. Wernhard, C.: Abduction in logic programming as second-order quantifier elimination. In: FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 103–119 (2013)
48. Wernhard, C.: Expressing view-based query processing and related approaches with second-order operators. Tech. Rep. KRR 14–02, TU Dresden (2014)
49. Wernhard, C.: Heinrich Behmann's contributions to second-order quantifier elimination. Tech. Rep. KRR 15–05, TU Dresden (2015)
50. Wernhard, C.: Second-order quantifier elimination on relational monadic formulas – A basic method and some less expected applications. In: TABLEAUX 2015. LNCS (LNAI), vol. 9323. Springer (2015)
51. Zach, R.: Completeness before Post: Bernays, Hilbert, and the development of propositional logic. Bulletin of Symbolic Logic 5, 331–366 (1999)