SCAN is complete for all Sahlqvist formulae*

Ullrich Hustadt
University of Liverpool, UK, ullrich@csc.liv.ac.uk

Introduction. One of the most general result on first-order definability and completeness in modal logic is Sahlqvist's [6] theorem where two notable facts are proved for a large, syntactically defined class of modal formulae, now called Sahlqvist formulae: first, the *correspondence result,* which says that Sahlqvist formulae all define first-order conditions on Kripke frames and these conditions can be effectively "computed" from the modal formulae; and second, the *completeness result*, which says that all those formulae are canonical, i.e. valid in their respective canonical frames, hence axiomatise completely the classes of frames satisfying the corresponding first-order conditions. The class of Sahlqvist formulae has been studied extensively, and several alternative proofs and generalisations have been proposed (see [3, 4]).

Computing the first-order equivalent (if it exists) of a modal formula amounts to the elimination of the universal monadic second-order quantifiers expressing the validity in a frame of that formula or, equivalently, the elimination of the existential monadic second-order quantifiers expressing the satisfiability of the formula. The best known algorithm for eliminating of existential second-order quantifiers is SCAN [2], though a number of alternative algorithms can be found in the literature [1, 7]. Of course, the algorithm does not work for all first-order definable modal formulae (at least because first-order definability of modal formulae is not decidable), but it has been proved correct whenever it gives an answer.

In this paper we show that SCAN successfully computes the first-order equivalents of all classical Sahlqvist formulae. To our knowledge, this is the first completeness result for an automated quantifier elimination procedure.

Sahlqvist formulae. We fix an arbitrary propositional (multi-)modal language. We assume that the primitive connectives in the language are $\neg$, $\wedge$, and the diamonds, while the others are definable as usual.

An occurrence of a propositional variable in a modal formula φ is *positive (negative)* iff it is in the scope of an even (odd) number of negations. A modal formula φ is *positive (negative) in a variable* q iff all occurrences of q in φ are positive (negative). A modal formula φ is *positive* (*negative*) iff all occurrences of propositional variables in φ are positive (negative). A *boxed atom* is a formula of the form $\Box_{k_1} \dots \Box_{k_n} q$, where $\Box_{k_1}, \dots, \Box_{k_n}$ is a (possibly empty) string of (possibly different) boxes and q is a propositional variable.

A *Sahlqvist antecedent* is a modal formula constructed from the propositional constants $\bot$ and $\top$, boxed atoms and negative formulae by applying $\vee$, $\wedge$, and diamonds. A *definite Sahlqvist antecedent* is a Sahlqvist antecedent obtained without applying $\vee$ (i.e. constructed from the propositional constants $\bot$ and $\top$, boxed atoms and negative formulae by applying only $\wedge$ and diamonds). A *(definite) Sahlqvist implication* is a modal formula $\varphi \to \psi$ where φ is a (definite) Sahlqvist antecedent and ψ is a positive formula. A *(definite) Sahlqvist formula* is a modal formula constructed from (definite) Sahlqvist implications by freely applying boxes and conjunctions, and by applying disjunctions to formulae without common propositional variables. A *basic Sahlqvist formula* is a definite Sahlqvist formula obtained from definite Sahlqvist implications without applying conjunctions.

It has been shown in [3] that every Sahlqvist implication is equivalent to a conjunction of definite Sahlqvist implications and every Sahlqvist formula is equivalent to a conjunction of basic Sahlqvist formulae.

The SCAN algorithm. SCAN reduces existentially quantified second-order sentences to equivalent first-order formulations. Given a second-order sentence containing existentially quantified second-order variables, the algorithm generates sufficiently many logical consequences, eventually keeping from the resulting set of formulae only those in which no second-order variables occur. The algorithm involves three stages:

(i) transformation to clausal form and Skolemization;
(ii) C-resolution;
(iii) reverse Skolemization (unskolemization).

The input of SCAN is a second-order formula of the form $\exists Q_1 \dots \exists Q_k\ \psi$, where the Q_i are existentially quantified unary predicate variables and ψ is a first-order formula. In the first stage SCAN converts ψ into clausal form, written $\mathrm{Cls}(\psi)$, by transformation into conjunctive normal form, inner Skolemization, and clausifying the Skolemized formula [5].

In the second stage SCAN performs a special kind of constraint resolution, called *C-resolution*. It generates all and only resolvents and factors with the second-order variables that are to be eliminated. In the process the algorithm produces equations and inequations of terms.

If the C-resolution stage terminates, this yields a set N of clauses in which the specified second-order variables are eliminated. This set is satisfiability-equivalent to the second-order formula originally given to the algorithm. If no clauses remain after purity deletion, i.e. $N = \emptyset$, then the original formula is a tautology; if C-resolution produces the empty clause, then it is unsatisfiable. If N is non-empty, finite and does not contain the empty clause, then in the third stage, SCAN attempts to reconstruct the quantifiers for the Skolem functions by reversing Skolemization. Reversing Skolemization is not always possible, for instance, it is not possible if the input formula is not first-order definable.

If the input formula is not first-order definable and stage two terminates successfully yielding a non-empty set not containing the empty clause then SCAN produces equivalent second-order formulae in which the specified second-order variables are eliminated but quantifiers involving functions

* Joint work with V. Goranko (Rand Afrikaans University, Johannesburg, South Africa), R. A. Schmidt (University of Manchester, UK), and D. Vakarelov (Sofia University, Bulgaria)

occur. These typically involve Henkin quantifiers. If SCAN terminates and the reverse Skolemization is successful, then the result is a first-order formula logically equivalent to the second-order input formula.

Since we intend to apply SCAN to Sahlqvist formulae, we now define a translation of modal formulae into second-order formulae. Let $\Pi(\varphi) = \forall Q_1 \ldots \forall Q_m \forall x \, \mathrm{ST}(\varphi, x)$, where $\mathrm{ST}(\varphi, x)$ is the (local) standard translation of a modal formula φ with a free variable x, and $Q_1, \ldots, Q_m$ are all the unary predicates occurring in $\mathrm{ST}(\varphi, x)$.

Let SCAN* denote the extension of the SCAN algorithm with the preprocessing and postprocessing necessary for computing first-order correspondences for modal logic formulae. The preprocessing involves translating the given modal formula to the second-order formula $\Pi(\varphi)$ and negating the result. The postprocessing involves negating the result output by SCAN, if it terminates.

Completeness of SCAN for Sahlqvist formulae. In the following, we show that SCAN* is complete for Sahlqvist formulae. We need to prove that for any second-order formula ψ obtained by preprocessing a Sahlqvist formula φ, SCAN can compute a first-order equivalent for ψ. To this end, one has to show two properties:

1. The computation of C-resolvents and C-factors terminates when applied to the set $\mathrm{Cls}(\psi)$ of clauses associated with ψ, i.e. SCAN can generate only finitely many new clauses in the process.
2. The resulting first-order formula, which in general contains Skolem functions, can be successfully unskolemized.

Since every Sahlqvist formula is equivalent to a conjunction of basic Sahlqvist formulae, it suffices to prove the result for every basic Sahlqvist formula φ.

We first consider the case that φ is a Sahlqvist implication. The cornerstone of our proof is the notion of a *chain*.

Let $(t_1, \ldots, t_n)$ be an ordered sequence of pairwise distinct terms. A *chain* C over $(t_1, \ldots, t_n)$ is a clause containing only literals of the form $(\neg)R_{k_i}(s, t)$ and $(\neg)Q_j(u)$ such that the following three conditions are satisfied:

(1) for every i, $1 \leq i \leq n{-}1$, either $\neg R_{k_i}(t_i, t_{i+1})$ or $R_{k_i}(t_i, t_{i+1})$ is in C;
(2) for every $(\neg)R_{k_i}(u, v) \in C$, $u = t_j$ and $v = t_{j+1}$ for some j, $1 \leq j \leq n - 1$;
(3) for every $(\neg)Q_j(u) \in C$, $u = t_j$ for some j, $1 \leq j \leq n$.

Lemma 1. *Let C be a chain over $(t_1, \ldots, t_n)$. Then there does not exist an ordered sequence $(s_1, \ldots, s_m)$ of pairwise distinct terms which is distinct from $(t_1, \ldots, t_n)$ such that C is also a chain over $(s_1, \ldots, s_m)$.*

The *length* of a chain C over $(t_1, \ldots, t_n)$ is n. Note that by Lemma 1 the chain C uniquely determines $(t_1, \ldots, t_n)$. So, the length of a chain is a well-defined notion.

Lemma 2. *Let Σ be a finite vocabulary of unary and binary predicate symbols, constant symbols and unary function symbols. If there is a bound l_b on the length of a chain and a bound d_b on the depth of terms that can occur in a chain, then the number of chains over Σ which are distinct modulo variable renaming is finitely bounded.*

Theorem 1. *Given any Sahlqvist implication φ,* SCAN* *effectively computes a first-order formula α_φ logically equivalent to φ.*

Proof (Sketch). Let N be the clausal form of $\mathrm{ST}(\neg\varphi, a)$. All the clauses that we can derive from N consist of (not necessarily variable-disjoint) chains over the vocabulary of N. The length of chains is bounded by the length of chains in N (modulo some small polynomial) and the depth of terms is bounded the depth of terms in N. We show that whenever we derive a clause with 'too many' chains in it, then it can be condensed. The specification of 'too many' uses the bound established in the proof of Lemma 2.

As to reverse Skolemization we can show by an inductive argument that the inequations and terms occurring in derived clauses can be successfully unskolemized. Consequently, restoration of the quantifiers can always be successfully accomplished during reverse Skolemization.

Finally, the general case of a basic Sahlqvist formula φ obtained from definite Sahlqvist implications by applying boxes, as well as disjunctions to formulae sharing no common variables can be dealt with by induction of the number of applications of such disjunctions. The basis of the induction (with no such applications) has been established by Theorem 1.

We can now state the main theorem.

Theorem 2 (Completeness with respect to Sahlqvist formulae). *Given any Sahlqvist formula φ,* SCAN* *effectively computes a first-order formula α_φ logically equivalent to φ.*

References

1. P. Doherty, W. Lukaszewics, and A. Szalas. Computing circumscription revisited: A reduction algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.
2. D. M. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992.
3. V. Goranko and D. Vakarelov. Sahlqvist formulas unleashed in polyadic modal languages. To appear in *Advances in Modal Logic*, vol. 3, Stanford University Press.
4. M. Kracht. How completeness and correspondence theory got married. In M. de Rijke, editor, *Diamonds and Defaults*, pages 175–214. Kluwer, 1993.
5. A. Nonnengart and C. Weidenbach. Computing small clause normal forms. In Robinson and Voronkov, editors, *Handbook of Automated Reasoning*, chapter 6, pages 335–367. Elsevier Science, 2001.
6. H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logics. In S. Kanger, editor, *Proc. of the 3rd Scandinavian Logic Symposium, 1973*, pages 110–143. North-Holland, 1975.
7. A. Szałas. On the correspondence between modal and classical logic: An automated approach. *Journal of Logic and Computation*, 3(6):605–620, 1993.