MAX-PLANCK-INSTITUT FÜR INFORMATIK

Functional Translation and Second-Order Frame Properties of Modal Logics

Hans Jürgen Ohlbach
Renate A. Schmidt

MPI–I–95–2–002 January 1995

INFORMATIK

Im Stadtwald
D 66123 Saarbrücken
Germany

Authors' Addresses

Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany.
Email: `(ohlbach,schmidt)@mpi-sb.mpg.de`.

Publication Notes

Submitted for publication to the *Journal of Logic and Computation.*

Acknowledgements

This work was supported by the ESPRIT Project 6471 MEDLAR and by the LOGO Project ITS 9102 funded by the BMFT. We thank U. Hustadt for reading the manuscript.

Abstract

Normal modal logics can be defined axiomatically as Hilbert systems, or semantically in terms of Kripke's possible worlds and accessibility relations. Unfortunately there are Hilbert axioms which do not have corresponding first-order properties for the accessibility relation. For these logics the standard semantics-based theorem proving techniques, in particular, the relational translation into first-order predicate logic, do not work.

There is an alternative translation, the so-called functional translation, in which the accessibility relations are replaced by certain terms which intuitively can be seen as functions mapping worlds to accessible worlds. In this paper we show that from a certain point of view this functional language is more expressive than the relational language, and that certain second-order frame properties can be mapped to first-order formulae expressed in the functional language. Moreover, we show how these formulae can be computed automatically from the Hilbert axioms. This extends the applicability of the functional translation method.

Keywords

Modal logic, functional semantics, transformation to many-sorted logic, correspondence problem, quantifier elimination, theorem proving for non-classical logics.

Contents

1 Motivation

From Minsky's early frame systems, which were defined purely operationally, and Brachman's KL-ONE knowledge representation system (Brachman and Schmolze 1985) to the language $\mathcal{ALC}$ of Schmidt-Schauß and Smolka's (1991) there has been a continuous trend in designing knowledge representation systems more and more according to logical principles with clear syntax and semantics and logical inferences as basic operations. $\mathcal{ALC}$ in particular is a language with the usual logical connectives $\wedge$, $\vee$, $\neg$ and the additional constructs `all(R C)` and `some(R C)`. For example, the following is an $\mathcal{ALC}$ definition which defines a 'concept' `proud-father` as a father all of whose children are successful persons.

$$\texttt{proud-father = father} \ \wedge \ \texttt{all(has-child successful-person)},$$

The fragment of $\mathcal{ALC}$ that includes the operations $\wedge$, $\vee$, $\neg$, `all`, `some` is just a variant of the multi-modal logic $\mathrm{K}_{(m)}$ (Schild 1991). `all(R C)` corresponds to $[R]C$ where the relational term `R` (a 'role' in KL-ONE jargon) is the parameter of the modal operator, and is interpreted as a binary accessibility relation.

$\mathcal{ALC}$ is still limited in its expressiveness. In pure $\mathcal{ALC}$ it is not possible to define concepts like, for a example, a city as a place with more than, say, 100 000 inhabitants. There are extensions of $\mathcal{ALC}$, like $\mathcal{ALCN}$, with additional operators, called 'number restrictions'.

$$\texttt{city = place} \ \wedge \ \texttt{atleast(100 000 inhabitant people)}$$

is a suitable $\mathcal{ALCN}$ definition. `atleast(`n` R C)` and `atmost(`n` R C)` restrict the number of so-called 'role fillers', i.e. they restrict the number of elements in the range of the relation `R` to $\geq n$ and $\leq n$, respectively.

The corresponding modal logic of $\mathcal{ALCN}$ is the multi-modal version of the system of 'graded modalities', which was introduced by Goble (1970) and Fine (1969, 1972) and which is investigated in Fattorosi-Barnaba and de Caro (1985), and van der Hoek (1992b, 1992a). Graded modalities are modal operators M_n for n a positive integer. $M_n\phi$ is true at a world x if there are *more* than n accessible worlds from x in which ϕ is true. This semantics is very natural and intuitive, but it has one disadvantage. All inference systems based on this semantics, in particular, tableaux systems, deal with these M_n-operators by generating a corresponding number of terms explicitly. For example, the formula $M_{100000}\mathit{people}$ triggers the generation of 100001 constant symbols as representatives for these objects. Except for counting these symbols and comparing the length of lists, there is no way to do arithmetic with these values. In particular there is no way of reasoning with symbolic arithmetic terms. For example, $M_{n+1}p \Rightarrow M_n p$ which is true for all n can only be verified in tableaux like systems for concrete instances, but not in general.

This is not the case for the Hilbert system axiomatizing the graded modalities. It is formulated with arithmetical terms, and in principle, this allows for invoking arithmetical computations. However, Hilbert systems have other disadvantages that makes them unsuitable to form the basis for automated reasoning. The natural relational semantics for graded modalities has limited value for doing theorem proving. We are investigating alternative semantics more suitable for building calculi, which may not capture our intuitions as well as the relational semantics does, but which are more suitable for automation. One idea we followed is to treat each M_n-operators as a standard multi-modal operator $[n]$, each one associated with an extra accessibility relation R_n. The frame structure, i.e. the set of worlds together with the accessibility relations, is axiomatized in such a way that the same theorems as in the original semantics hold. These axioms would be formulated with terms of arithmetic such that arithmetical calculations replace the counting the symbols.

Unfortunately it turns out that the axioms defining the appropriate frame structure have a second-order nature, and reasoning in second-order predicate logic is extremely difficult. Thus, adding some very simple means for doing arithmetical calculations into our knowledge representation formalism left us in second-order predicate logic. A closer analysis of the frame axioms (the determining frame conditions), however, reveals that the second-order aspect does not arise from the frame structure itself, but from the limited expressiveness of the relational language of accessibility relations. Changing the language in such a way that the underlying semantic structures are retained solves the problem.

Every relation R can be reformulated as a set AF of 'accessibility functions':

$$R(x,y) \Leftrightarrow \exists f \in AF\ y = f(x).$$

In an enriched language that includes a distinguished *apply* function symbol, literals $R(x,y)$ can therefore be replaced by terms $apply(f,x)$. The information about accessible worlds is then encoded in the structure or the terms built with the *apply*-function. It turns out that this *functional language* is in some sense more fine grained than the relational language, which makes it easier to express complex properties of accessibility relations.

In this paper we do not consider the graded modalities. We investigate the transformation of second-order frame properties formulated in the relational language to (possibly first-order) properties formulated in the functional language. We show how the 'functional' frame properties can be computed automatically from the Hilbert axioms of the given modal system.

2 Introduction

The modal logics we are primarily concerned with in this paper are propositional modal logics above the system K. The logics we consider are normal modal logics which are extensions of standard propositional logic with the two modal operators $\Box$ and $\Diamond$. These systems are defined by the axioms for propositional logic together with the rule of modus ponens, the axiom K, $\Box(p \Rightarrow q) \Rightarrow (\Box p \Rightarrow \Box q)$ and the rule of *necessitation* 'from p infer $\Box p$', and arbitrary additional (modal) axioms. The Lindenbaum construction applied to these systems yields (subclasses of the class of) *Boolean Algebras with operators* for which Jónsson and Tarski (1951) proved representation theorems. They showed that every Boolean Algebra can be embedded in a Boolean set algebra with operators defined set-theoretically. This means the elements of an algebra can be interpreted as sets and the extra n-ary operators can be interpreted as $(n+1)$-ary relations. Only later, Kanger and then Kripke (1959, 1963) proved the analogous completeness result for the modal logics using non-algebraic constructions. The semantic structures they used became known as Kripke frames and Kripke models (frames correspond to atom structures of complex algebras in Jónsson and Tarski (1951)). K is the weakest normal modal logic[1].

A *frame* for a normal modal system is a pair (W, R) of a (non-empty) set W of *worlds* and a binary relation R over W, called the *accessibility relation*. An *interpretation* $\Im$ (also called *model*) for a normal modal system consists of a frame and an *assignment* of sets of worlds to the propositional variables. A propositional variable p is said to be *true* at a world w in $\Im$, written $w \models p$, if w is in the set of worlds assigned to p. A formula $\Box\varphi$ is true at a world w if φ is true at *all* worlds R-accessible from w. The dual formula $\Diamond\varphi$ is true at a world w if φ is true at *some* world R-accessible from w. The semantics of the classical connectives $\wedge$, $\vee$, $\neg$ etc. is as usual. A modal formula is *valid in a frame* iff *for all assignments* it is true *at all the worlds* of the frame. According to this definition, the K-axiom is valid in all possible frames. Semantically the necessitation rule is interpreted by the following: 'if p is true at all worlds in a frame then $\Box p$ is also true at all worlds of the frame'. Accordingly, the necessitation rule is valid, too, in all possible frames. In fact, the K-axiom and the necessitation rule are the characterizing axioms and rules for the possible worlds semantics with binary accessibility relations. For weaker non-normal modal systems other weaker semantic structures are needed (Chellas 1980). Since we focus exclusively on normal systems we assume the K-axiom and the necessitation rule to be always valid in the standard Kripke semantics.

Modal systems with additional modal axioms other than the K-axiom and the necessitation rule are not necessarily characterized by all possible frames. The additional (stronger) axioms are not valid in all possible frames, though they may be valid in all frames of a certain subclass of frames. For example, the axiom T, $\Box p \Rightarrow p$, is valid in all frames in which the accessibility relation is *reflexive*. This means, reflexivity is the characteristic frame property for the system T (that is, the system K extended with the axiom T). Not every axiom can be reduced to a characteristic frame property. Some extensions of K cannot be associated with classes of characteristic frames, but they can be associated with classes of characteristic interpretations, i.e. frames together with assignments (these are often called extended frames). An example of such an axiom is the axiom VB (short for 'van Benthem') $\Diamond\Box p \vee \Box(\Box(\Box q \Rightarrow q) \Rightarrow q)$ (Hughes and Cresswell 1984, page 57). In this paper we will not be interested in such *incomplete* systems.

For devising inference methods based on the possible worlds semantics it is important to know the characteristic frame property for a given Hilbert axiom. It is even better if this frame property is definable in first-order logic. There are different methods for finding the characteristic frame properties for Hilbert axioms. The Sahlquist-van Benthem technique is the most widely known method (Sahlqvist 1975, van Benthem 1984). Other methods developed more recently are by Szałas (1992), Simmons (1994), Ohlbach and Gabbay (1992), and Ohlbach (1994). The approach of Ohlbach and Gabbay exploits that all modal axioms can be translated to *second-order predicate logic* (Ohlbach 1995). For example, the axiom $\Box p \Rightarrow p$ translates to

$$\forall p \forall w \; sat(w, \Box(p)) \Rightarrow sat(w, p).$$

[1] Non-normal modal systems are without the K-axiom or the necessitation rule (Chellas 1980).

Based on the semantics of $\Box$, this in turn translates to

$$\forall p \forall w (\forall v \; R(w, v) \Rightarrow p(v)) \Rightarrow p(w)). \tag{1}$$

Given the Kripke semantics for the modal operators, this *relational translation method* seems natural and easy, but the proof that the transformation of Hilbert axioms into second-order predicate logic works, that is, that the theorems are preserved, is rather involved (Ohlbach 1994, Ohlbach 1995).

The *correspondence problem* now amounts to finding a formula which is equivalent to the second-order formula, but does not contain the predicate variables, just the binary predicate symbol R and possibly the equality symbol. For T the corresponding first-order formula equivalent to (1) exists, namely $\forall w \; R(w, w)$. But it may also be the case that the language of first-order logic is expressively too weak for describing certain frame properties in terms of the R-symbol. In this paper we address the problem:

> *For modal axioms, for which equivalent first-order formulae do not exist in general, can we find a first-order formulation in a different language more expressive than the relational language?*

In Gabbay and Ohlbach (1992) a quantifier elimination algorithm called SCAN is presented which was developed for the purpose of finding the equivalent formula α' for any given second-order formula α[2]. The SCAN algorithm can be used to point out one source of the problem. There is one particular step where the algorithm fails to compute a first-order formula, but a slight modification on the input formulae makes it succeed.

In the following we give a brief description of the SCAN algorithm and show what kind of manipulation of the input formula is necessary.

Quantifier elimination with SCAN

The SCAN algorithm is based on an idea which appeared already in Ackermann (1935, 1954). SCAN takes as input only second-order formulae of the form

$$\alpha = \exists p_1, \ldots, \exists p_k \; \psi$$

with existentially quantified predicate variables p_i and ψ a first-order formula. If the predicate variables of the formula under consideration are universally quantified, we negate the formula first, which turns the universal quantifiers into existential quantifiers, we apply SCAN, and negate SCAN's result.

SCAN performs the following three steps:

1. ψ is transformed into clause form.

[2] SCAN is not complete for arbitrary second-order formulae. By a *complete* quantifier elimination algorithm we mean an algorithm that is guaranteed to find the equivalent first-order formula if there is one. In general, no complete algorithm exists, for otherwise arithmetic is enumerable. For the particular case of second-order formulae arising from the relational translation of modal axioms, it is an open problem whether a complete algorithm exists. For the purpose of this paper this is not relevant.

2. All C-resolvents and C-factors with the predicate variables $p_1, \ldots, p_n$ are generated. *C-resolution* ('C' is short for constraint) is defined as follows:

$$\begin{array}{ll} p(s_1,\ldots,s_n) \vee C & p(\ldots) \text{ and } \neg p(\ldots) \\ \neg p(t_1,\ldots,t_n) \vee D & \text{are the } \textit{resolution literals} \\ \hline C \vee D \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n & \end{array}$$

and the *C-factorization* rule is defined analogously:

$$\frac{p(s_1,\ldots,s_n) \vee p(t_1,\ldots,t_n) \vee C}{p(s_1,\ldots,s_n) \vee C \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n}.$$

When *all* resolvents and factors between a particular literal and the rest of the clause set have been generated (the literal is said to be 'resolved away'), the clause containing this literal is deleted (this is called purity deletion). If all clauses have been deleted this way, we know α is a tautology. If an empty clause is generated, we know α is contradictory.

3. If step 2 terminates and the set of clauses is non-empty then reconstruct the quantifiers for the Skolem functions.

Take, for example, the reflexivity formula (1). Since (1) contains universally quantified predicate variables, we negate it first to get existentially quantified predicate variables. The negation is

$$\exists p \exists w\ (\forall v\ R(w,v) \Rightarrow p(v)) \wedge \neg p(w)),$$

in clause form:

$$\begin{array}{l} \neg R(a_w, v), p(v) \\ \neg p(a_w). \end{array}$$

a_w is the Skolem constant for w. The only C-resolution step possible with p-literals yields $\neg R(a_w, a_w)$. Reversing the Skolemization process we reintroduce quantifiers and get $\exists w\ \neg R(w,w)$. This is negated again and the final result is $\forall w\ R(w,w)$.

There are two critical steps in the algorithm: the resolution may not terminate, and the reconstruction of the quantifiers for the Skolem functions may not be possible. Resolution loops, for example, if we apply SCAN to the axiom G, $\Diamond p \Rightarrow \Diamond(p \wedge \Box\neg p)$, which determines frames with finite R-chains, and this is not a first-order property.

In this paper we investigate the second critical step, the impossibility to reconstruct the quantifiers for the Skolem functions. For the purpose of computing frame properties for Hilbert axioms we show what kind of manipulations of the input formula for SCAN are possible such that this problem does not occur any more. The simplest modal axiom for which our manipulation works is the *McKinsey axiom*

$$M \quad \Box\Diamond p \Rightarrow \Diamond\Box p. \tag{2}$$

(The critical axioms in graded modal logics, where we first encountered the problem, have a very similar structure.) The negation of the transformed McKinsey axiom

$$\exists p \exists w\ (\forall u\ R(w,u) \Rightarrow (\exists a\ R(u,a) \wedge p(a))) \wedge (\forall v\ R(w,v) \Rightarrow (\exists b\ R(v,b) \wedge \neg p(b))) \tag{3}$$

is converted into clause form

$$\begin{array}{lll} C_1 & \neg R(c_w, u), R(u, f_a(u)) & f_a \text{ is the Skolem function for } a \\ C_2 & \neg R(c_w, u), p(f_a(u)) & c_w \text{ is the Skolem constant for } w \\ C_3 & \neg R(c_w, v), R(u, g_b(v)) & g_b \text{ is the Skolem function for } b \\ C_4 & \neg R(c_w, v), \neg p(g_b(v)) & \end{array}$$

Again, there is only one C-resolution step possible with p, namely between C_2 and C_4. The resulting clause set is

$$\begin{array}{ll} C_1 & \neg R(c_w, u), R(u, f_a(u)) \\ C_3 & \neg R(c_w, v), R(u, g_b(v)) \\ C_5 & \neg R(c_w, u), \neg R(c_w, v), f_a(u) \neq g_b(v) \end{array}$$

It is not possible to reconstruct the existential quantifiers for the Skolem functions f_a and g_b[3]. The problem is that f_a depends on u and g_b depends on v and both $f_a(u)$ and $g_b(v)$ occur in the same clause (C_5). If we were allowed to change the variable dependency for f_a and g_b in a suitable way *unskolemization* would be possible. Changing the variable dependency means, moving in (3) the existential quantifiers in front of the universal quantifiers:

$$\exists p \exists w\ (\exists a \forall u\ R(w, u) \Rightarrow (R(u, a) \wedge p(a))) \wedge (\exists b \forall v\ R(w, v) \Rightarrow (R(v, b) \wedge \neg p(b))).$$

If we apply C-resolution to this formula we get

$$\begin{array}{ll} C_1 & \neg R(c_w, u), R(u, f_a) \\ C_3 & \neg R(c_w, v), R(u, g_b) \\ C_5 & \neg R(c_w, u), \neg R(c_w, v), f_a \neq g_b \end{array}$$

which can be unskolemized to a linearly quantified (first-order) formula which we can negate without problems. But, of course, moving existential quantifiers outward, and in particular, over universal quantifiers, is in general not admissible. This transformation is not equivalence preserving. We will show that in a functional language (which we'll use to replace the relational language) moving existential quantifiers outward over universal quantifiers is allowed in certain cases. The idea for this was first mentioned by Herzig and others (Fariñas del Cerro, Luis and Herzig 1988a, Herzig 1989).

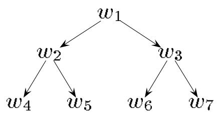
The functional language

Every relation can be decomposed into a set of functions. The accessibility relation can therefore be decomposed into a set AF of *accessibility functions* mapping worlds to accessible worlds. Consider the relation R_1:

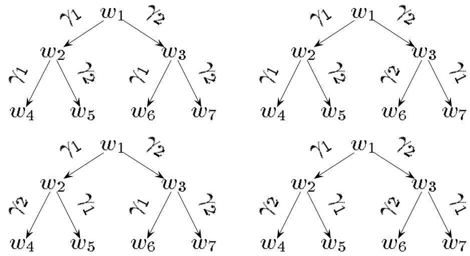
[3] There is a way of reconstructing quantifiers for f_a and g_b by means of parallel *Henkin Quantifiers*.

$$\exists c \begin{pmatrix} \forall u \exists x \\ \forall v \exists y \end{pmatrix} \begin{array}{l} (R(c, u) \Rightarrow R(u, x)) \quad \wedge \quad (R(c, v) \Rightarrow R(v, y)) \\ \wedge \quad ((R(c, u) \wedge R(c, v)) \Rightarrow x \neq y). \end{array}$$

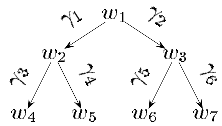
Unfortunately Henkin quantifiers are essentially second-order in nature. Since the formula must be negated in order to get the frame property for the McKinsey axiom, this form is useless for the purpose of automated reasoning.



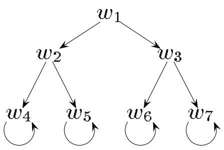
There are at least four different ways to decompose R_1 into sets of two functions $\{\gamma_1, \gamma_2\}$.



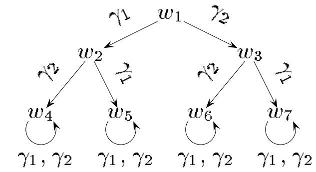
Since the relation is not serial, this means, there are dead ends, all these γ_1's and γ_2's are partial functions. With partial functions there are even more possible decompositions for a relation. The picture below shows an extreme case where the functions $\gamma_1, \ldots, \gamma_6$ are 'as partial as possible'.



For total (serial) relations it is always possible to decompose the relation into a set of *total* functions. For example, the relation R_2:

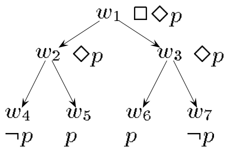


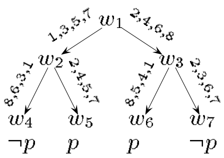
can be decomposed into



and the γ_i are all total. This is an important observation for our purposes.

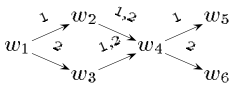
Also note, there are many *different* 'functional frames', that is, sets of accessibility functions, which represent the *same* relational frame. Observe, one such functional frame is sufficient for proving the existence of a model. This means, we have some freedom in choosing a functional frame which is suitable for our purposes (which is normalized in a certain sense). We are interested in those functional frames that justify moving existential quantifiers over universal quantifiers. Let us illustrate the basic idea with an example. Suppose a formula $\Box\Diamond p$ is true at the world w_1, i.e. suppose $w_1 \models \Box\Diamond p$, in our first frame $(\{w_1, \ldots, w_7\}, R_1)$. For every world w accessible from w_1 there is a world accessible from w where p is true, i.e. $\forall w\ R(w_1, w) \Rightarrow \exists v\ R(w, v) \wedge v \models p$ holds for R_1. Suppose the situation as in the left picture below. In this model the formula $\exists v \forall w\ R(w_1, w) \Rightarrow\ R(w, v) \wedge v \models p$ in which we have swapped the existential quantifier $\exists v$ with the universal quantifier $\forall w$ is false.



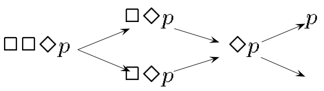


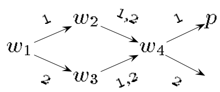
But now consider the functional frame in the right picture. It combines the four frames depicted above. The numeric labels i denote the accessibility functions γ_i. In the functional language we can express the fact that $\Box\Diamond p$ is true at w_1 by $\forall\gamma\exists\delta\ \delta(\gamma(w_1)) \models p$. For this model we can swap the $\exists\delta$ quantifier and the $\forall\gamma$ quantifiers. $\exists\delta\forall\gamma\delta(\gamma(w_1)) \models p$ is also true at w_1, because the function γ_4 (as well as the function γ_5) maps the worlds w_2 and w_3 to a world where p holds. Moreover, regardless in which one of the worlds w_4, w_5, w_6 or w_7 p is true, in this model there is always a function γ_i which maps w_2 and w_3 to the right worlds. We will show that in functional frames which are *maximal* as this example is, it is always justified to move the existential quantifiers in front of the universal quantifiers. By maximal functional frames we mean those in which the set of accessibility functions contains all possible accessibility functions. Every relational frame can always be associated with a maximal functional frame, and this is what we need.

Our sample frame has the form of a tree. For tree structures the decomposition into function sets is easier than for non-tree-like structures. Consider the frame



and suppose $\Box\Box\Diamond p$ is true at w_1. For all worlds u accessible from w_1 (w_2 and w_3) and all worlds v accessible from u (w_4) there is a world z accessible from v such that p is true at z. The existentially quantified z depends on u and v. In this frame there is only one possible instance for v (namely w_4). Therefore, z depends actually only on u. The dependencies can be as follows: for $u = w_2$ assign w_5 to z and for $u = w_3$ assign w_6 to z (or vice versa). Different paths for reaching $v = w_4$ may continue to different worlds. Fortunately in propositional modal logics this situation does not arise. If one path crossing w_4 is extended to a world satisfying some p then all other paths crossing w_4 can be extended to the same world. To illustrate this, consider again the formula $\Box\Box\Diamond p$ which is true at w_1. Then $\Box\Diamond p$ is true at w_2 and w_3 and $\Diamond p$ is true at w_4. It is sufficient that p is true at either w_5 or w_6, not necessarily in both worlds.





The left picture is a model for the formula $\Box\Box\Diamond p$. In the corresponding functional frame (depicted on the right) $\forall\gamma\gamma'\exists\delta\ \delta(\gamma'\gamma(w_1)) \models p$ is true. The assignment of γ_1 to δ allows us to move the $\exists\delta$ quantifier in front of $\forall\gamma\gamma'$. In this model $\exists\delta\forall\gamma\gamma'\ \delta(\gamma'\gamma(w_1)) \models p$ is satisfied. We have exploited the fact that in propositional modal logics, a statement that a formula is true at a particular world depends only on this world and the worlds accessible from this one, not on the path we took to reach the current world.

This is not true in quantified modal logic. An example (due to A. Herzig) shows what can happen. The formula $\Box(\exists x\ (p(x) \wedge \Box\Diamond\neg p(x)))$ is true at the world w_1 of the following model.

$$\begin{tikzcd}[row sep=small]
 & w_2 \quad p(a), \neg p(b) \\
w_1 \arrow[ur, leftrightarrow] \arrow[dr, leftrightarrow] & \\
 & w_3 \quad p(b), \neg p(a)
\end{tikzcd}$$

For every world u accessible from w_1 (these are w_2 and w_3) there is a world x in which $p(x)$ holds (for w_2, x is a and for w_3, x is b), and for every worlds v accessible from u (w_1) there is a world y accessible from v (w_2 and w_3 are the candidates) such that $\neg p(x)$ holds at y. Now we have to choose either w_2 or w_3 and check whether $\neg p(x)$ holds, but x was determined in a previous world, in case that $u = w_2$, x is a and in case that $u = w_3$, x is b. Our choice depends on the path we choose to get to $v = w_1$. This example shows that we must be careful where we apply the trick of moving existential quantifiers to front. For quantified modal logic the trick does not work in general, it doesn't for relational frames and it doesn't for functional frames. In this paper we restrict our attention to the propositional case.

In the technical part of the paper we proceed as follows: We briefly recall how modal formulae can be translated into (first-order or second-order) predicate logic based on their relational semantics. We show how this relational translation can be transformed according to the functional translation (Ohlbach 1988a, Fariñas del Cerro, Luis and Herzig 1988b, Herzig 1989, Auffray and Enjalbert 1992, Zamov 1989). Then we prove a theorem that justifies the application of the rule for moving existential quantifiers outward over universal quantifiers. Finally, we show how the SCAN algorithm can be used to compute the frame properties in the functional language.

3 Translation from modal into predicate logic

The form of specification of modal systems we are interested in are Hilbert axiomatizations. Hilbert axiomatizations provide a means for specifying a logic purely syntactically, without any reference to a model theoretic semantics.

Recall, the basic definitions. The set of (propositional) modal formulae over an infinite set of propositional variables is defined inductively to be the least set such that every propositional variable is a modal formula, and if p and q are modal formulae, then $p \wedge q$, $p \vee q$, $p \Rightarrow q$, $p \Leftrightarrow q$, $\neg p$, $\Box p$ and $\Diamond p$ are modal formulae. A *Hilbert axiom* is just a modal formula. A *Hilbert rule* is a rule of the form 'from A_1 and ... and A_n derive A', where the A_i and A are modal formulae. A *Hilbert system* is given by a set of Hilbert axioms and Hilbert rules. An instance A' of a Hilbert axiom or rule A is an axiom or rule $A[\vec{p}/\vec{\varphi}]$ where all occurrences of the propositional variables p_i are replaced with the modal formula φ_i. A modal formula φ is a Φ-*theorem* for a set Φ of Hilbert axioms and rules iff φ can be derived from Φ by applying instances of the Hilbert rules to instances of the Hilbert axioms.

The Kripke semantics for the modal operators forms the basis for the *relational translation* of modal formulae, and in particular, of Hilbert axioms and rules, into predicate logic. Propositional variables become unary predicate symbols, whose argument variables denote a world in which the predicate is to be evaluated. Since propositional variables of Hilbert axioms and rules are actually place holders for formulae, they become predicate variables. The semantics of the modal operators and the other connectives are used as translation rules.

Definition 1 (Relational translation) The *relational translation function* Π_r converts Hilbert axioms and rules, and the theorem to be proved to second-order predicate logic formulae. As an auxiliary function we use π_r that maps tuples of modal formulae and 'world variables' to second-order expressions.

$$\begin{array}{lcll} \pi_r(p, w) & = & p(w) & \text{if } p \text{ is a predicate symbol} \\ \pi_r(\Box\psi, w) & = & \forall v\ R(w, v) \Rightarrow \pi_r(\psi, v) & \\ \pi_r(\Diamond\psi, w) & = & \exists v\ R(w, v) \wedge \pi_r(\psi, v) & \end{array}$$

and for the propositional connectives π_r is a homomorphism.

A *relational translation* of a modal formula A with formula variables $p_1, \ldots, p_n$ is defined by

$$\Pi_r(A) \stackrel{\text{def}}{=} \forall p_1, \ldots, p_n \forall w\ \pi_r(A, w)$$

For a Hilbert rule of the form 'from A_1 and ... and A_n infer A'

$$\Pi_r(A_1) \wedge \ldots \wedge \Pi_r(A_n) \Rightarrow \Pi_r(A)$$

is the relational translation. Π_r maps a set Φ of Hilbert axioms and rules to the conjunction of the relational translation of the members:

$$\Pi_r(\Phi) = \bigwedge_{A \in \Phi} \Pi_r(A).$$

$\lhd$

The translation with Π_r of Hilbert axioms and rules yields formulae in a fragment of higher-order predicate logic for which we need a system that includes apart from the first-order machinery also an extension that accommodates quantification over (one-place) predicate variables. The semantics of this logic is the natural extension of the Tarskian semantics for first-order logic where the assignment part of an interpretation $\Im$ not only assigns domain elements to domain variables, but also n-ary relations over the domain to n-place predicate symbols.

The following soundness and completeness result for the relational translation is proved in Ohlbach (1994, 1995). It extends Jónsson and Tarski's representation theorem for Boolean Algebras with operators as well as the Kripke semantics for modal logic by giving a representation for arbitrary instances of these systems.

Theorem 2 (Soundness and completeness of the relational translation) Let φ be a modal formula. The following are equivalent statements:

(i) A modal formula φ is a Φ-theorem.
(ii) $\Pi_r(\Phi) \Rightarrow \Pi_r(\varphi)$ is a predicate logic theorem.
(iii) $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ is inconsistent[4]. $\lhd$

[4] In this paper the notion of consistency is understood to be a semantic notion. By 'A is consistent' we mean formula A is satisfiable in some model. Since predicate logic (and also many-sorted predicate logic) is complete, A is inconsistent (or unsatisfiable) iff $\vdash \neg A$.

Various results, in particular the Sahlquist-theorem, relate Hilbert axioms of a special syntactic form to their corresponding frame properties (van Benthem 1984). In contrast to this approach, we view the derivation of correspondence properties for Hilbert axioms as a two-step process. First, we convert a Hilbert axiom A to the second-order formula $\Pi_r(A)$, and second, to a formula free of second-order predicate variables determining the characterizing frame structures.

Note that the Hilbert axioms and rules for propositional logic as well as the K-axiom and the necessitation rule are tautologies in the relational translation. It is sufficient, therefore, to consider only the extra modal axioms. Thus, from here on we may assume Φ is the set containing just the extra axioms which define an extension of K.

4 Translation into the functional language

The functional language can be introduced in different ways. Most authors choose to define a direct translation from the modal language into a predicate logic language having special terms denoting accessibility functions. The behaviour of arbitrary modal axioms has to our knowledge never been studied in this functional context. In this paper we use a different translation approach which was first proposed in Ohlbach, Gabbay and Plaisted (1994). This approach is in certain ways simpler and we hope it clarifies the treatment of the Hilbert systems under consideration. We start with the relational translation of modal formulae. We want to replace the accessibility relation symbol R by terms with accessibility functions. To this end we add to $\Pi_r(\Phi)$ (recall, Φ defines a normal modal logic) the following definition of the accessibility relation in terms of the accessibility functions:

$$R(x, y) \Leftrightarrow \exists\gamma\ y = \gamma(x). \tag{4}$$

We then use this equivalence for replacing all occurrences of R by an instance of the right hand side of (4). In general, adding another formula to a set of formulae may introduce inconsistencies. We have to make sure that this does not happen. This means we must prove that,

$$\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi) \text{ is consistent} \quad \text{iff} \quad (4) \wedge \Pi_r(\Phi) \wedge \neg\Pi_r(\varphi) \text{ is consistent.}$$

The '$\Leftarrow$'-part of this proof is trivial, since removing a formula does not introduce inconsistencies. The '$\Rightarrow$'-part of the proof, which is given below, requires the construction of an appropriate set of accessibility functions, so that any model of $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ can be extended to a model that satisfies (4) as well. This is the part of the proof where we exploit the freedom to construct not an arbitrary 'functional model', but a functional model satisfying the maximality condition we described in Section 2, which justifies, us moving existential quantifiers outward over universal quantifiers.

For technical and also for presentation reasons, we find it useful to use a sorted logic as target language for the functional translation. We employ the basic many-sorted logic with a sort hierarchy and single sort declarations for function symbols (Walther 1987). In this logic, a sort symbol can be viewed as a unary predicate and denotes a subset of the domain of interpretation. A subsort declaration $S \sqsubseteq T$ is interpreted as a subset relation $\Im(S) \subseteq \Im(T)$, and a function sort declaration like, for example, $f : S_1 \times S_2 \to S$ is interpreted as $\forall x, y\ S_1(x) \wedge S_2(y) \Rightarrow S(f(x,y))$. Quantification can restrict variables to a sort. $\forall x{:}S\ \psi$ is interpreted as $\forall x\ S(x) \Rightarrow \psi$ and $\exists x{:}S\ \psi$ is interpreted as $\exists x\ S(x) \wedge \psi$. Any unsorted logic can be embedded in a sorted logic by introducing just one single sort denoting the entire domain. The relational translation is embedded in the sorted context by introducing the sort W (for worlds) and consider $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ to be a formula of sorted logic.

One further sort symbol, AF, which denotes the sort for accessibility functions is necessary. The accessibility functions are variables over which we quantify. We do not want such variables occuring in function symbol places and introduce a special function symbol $\downarrow$ that will enable us to write $\downarrow(\gamma, w)$ instead of $\gamma(w)$ for γ an accessibility function symbol.

Now we prove the basic soundness and completeness theorem for the functional translation. We prove it first for the case of logics defined by serial frames. Recall, these are logics extending KD. The non-serial case is technically a bit more complicated, but brings no new insight.

Theorem 3 (Functional extension) Let Φ be a Hilbert system extending KD. That is, Φ includes $\Box p \Rightarrow \Diamond p$. A modal formula φ is a Φ-theorem iff

$$\Psi \wedge \Pi_r(\Phi) \wedge \neg\Pi_r(\varphi) \tag{5}$$

is inconsistent, where Ψ is the set containing the following sort declaration and axiom:

$$\downarrow: AF \times W, \to W \tag{6}$$

$$\forall x, y{:}W\ \ R(x,y) \Leftrightarrow \exists \gamma{:}AF\ y = \downarrow(\gamma, x) \tag{7}$$

Proof By Theorem 2, φ is a Φ-theorem iff $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ is inconsistent. It suffices to show that every model $\Im$ of $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ can be extended to a model $\Im'$ of both $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ and Ψ.

Definition 4 Given a relational model $\Im$ we define its *functional extension* to be the model $\Im' \stackrel{\text{def}}{=} \Im(AF, \downarrow)$ with:

(i) The interpretations of W and R in $\Im'$ are the same as in $\Im$,
(ii) $AF^{\Im'} \stackrel{\text{def}}{=} \{\gamma : W^{\Im} \to W^{\Im} \mid \gamma \text{ is total and } \forall w \in W^{\Im}\ R^{\Im}(w, \gamma(w)) \text{ is true}\}$, and
(iii) $\downarrow^{\Im'}$ is the application function, meaning $\downarrow^{\Im'}(\gamma, x) = \gamma(x)$. $\lhd$

Obviously, this definition satisfies the sort declaration (6). That the '$\Leftarrow$'-direction of (7) holds is also obvious. The '$\Rightarrow$'-direction follows from the fact that $AF^{\Im'}$ consists of *all possible total* functions mapping worlds to R-accessible worlds and $AF^{\Im'}$ is non-empty. This completes the proof of Theorem 3. $\lhd$

The functional extension $\Im'$ of $\Im$ constructed in Theorem 3 (Definition 4) is conservative in the sense that it does not assign different meanings to the symbols for which $\Im$ is defined, in particular, to the symbols R and W, and it assigns a meaning to the new symbols AF and $\downarrow$. We can define $\Im'$ for arbitrary relational models with serial accessibility relation. This has some very useful side effects. We can add to (5) other declarations and formulae provided they are true in the functional extension $\Im'$, or, there is another extension of $\Im'$ which makes them true. Thus, we can state the following corollary:

Corollary 5 Let Φ be a Hilbert system extending KD, let Ψ be defined as in Theorem 3, and let Ψ' be a set of R-free sort declarations and formulae. A modal formula φ is a Φ-theorem iff

$$\Psi \wedge \Pi_r(\Phi) \wedge \neg\Pi_r(\varphi) \wedge \Psi'$$

is inconsistent, provided the declarations and formulae in Ψ' hold in the functional extension $\Im'$ (defined in Definition 4) or $\Im'$ has an extension that satisfies Ψ'.

Proof Proceed as in the previous theorem. The 'if' direction is evidently true. For the 'only if' direction it suffices to show that every model $\Im$ of $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ can be extended to a model $\Im''$ of $\Psi \wedge \Pi_r(\Phi) \wedge \neg\Pi_r(\varphi) \wedge \Psi'$. Construct from $\Im$ the functional extension $\Im'$. $\Psi \wedge \Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$, that is, (5), is true in $\Im'$. If $\Im'$ satisfies Ψ' then choose $\Im''$ to be $\Im'$. If it does not, choose $\Im''$ to be the extension of $\Im'$ which is required to exist and which satisfies (5) $\wedge\ \Psi'$. $\lhd$

In (7) we have a definition of the accessibility relation R in terms of the sort AF. Since this is an equivalence, the process of rewriting all occurrences $R(s,t)$ with the appropriate instances of the right hand side of (7) preserves equivalence. Suppose we do this, i.e. we eliminate in $\Pi_r(\Phi)$ and in $\Pi_r(\varphi)$ all occurrences of R and replace them by expressions in terms of the accessibility functions. We end up with formulae $\Pi^*_r(\Phi)$ and $\Pi^*_r(\varphi)$, together with the declaration (6) and axiom (7). But this is not the end. Nothing would have been gained if we keep the formula (7) in our formula set because we cannot prevent a theorem prover trying to prove $\Pi^*_r(\Phi) \Rightarrow \Pi^*_r(\varphi)$ from using (7) in the '$\Leftarrow$'-direction, thus, restoring old R-formulae. We want to delete axiom (7) after having done the rewriting step. Deleting an axiom may turn an inconsistent set of formulae into a consistent set. In this case, fortunately this cannot happen. Proving that every model for the transformed formula set without (7) can be turned into a model with (7) is very easy. After all occurrences of R have been rewritten, (7) is the only formula remaining in which R occurs. This is also true, if, according to Corollary 5 more formulae Ψ' are added, because these additional formulae are assumed to be R-free. This means we can just use the equivalence (7) as a definition for R. Every model for $\Pi^*_r(\Phi) \wedge \neg\Pi^*_r(\varphi) \wedge \Psi'$, *regardless whether the sort AF is interpreted as a set of functions and the $\downarrow$-symbol is interpreted as 'apply'-function or not*, can therefore be extended by an interpretation for the symbol R such that (7) is satisfied.

The relational translation of the modal operators yields formulae with a characteristic syntactic pattern. We can exploit these for combining the relational translation Π_r with the rewriting step that eliminates R into one single transformation. For the $\Box$-operator it is $\pi_r(\Box\psi, w) = \forall u\ R(w,u) \Rightarrow \pi_r(\psi, u)$ and for the $\Diamond$-operator it is $\pi_r(\Diamond\psi, w) = \exists u\ R(w,u) \wedge \pi_r(\psi, u)$. Eliminating R by using (7) as rewrite rule yields

$$\begin{array}{rcl}
\pi_r(\Box\psi, w) & = & \forall u\ R(w,u) \Rightarrow \pi_r(\psi, u) \\
 & \xrightarrow{*} & \forall u\ ((\exists\gamma{:}AF\ u = \downarrow(\gamma, w)) \Rightarrow \pi_r(\psi, u)) \\
 & \Leftrightarrow & \forall u\ \forall\gamma{:}AF\ (u = \downarrow(\gamma, w) \Rightarrow \pi_r(\psi, u)) \\
 & \Leftrightarrow & \forall u\ \forall\gamma{:}AF\ \pi_r(\psi, \downarrow(\gamma, w)) \\
 & \Leftrightarrow & \forall\gamma{:}AF\ \pi_r(\psi, \downarrow(\gamma, w))
\end{array}$$

and

$$\begin{array}{rcl}\pi_r(\Diamond\psi, w) & = & \exists u\ R(w,u) \wedge \pi_r(\psi, u) \\ & \xrightarrow{*} & \exists u\ (\exists\gamma{:}AF\ u = \downarrow(\gamma, w)) \wedge \pi_r(\psi, u) \\ & \Leftrightarrow & \exists u\ \exists\gamma{:}AF\ (u = \downarrow(\gamma, w) \wedge \pi_r(\psi, u)) \\ & \Leftrightarrow & \exists u\ \exists\gamma{:}AF\ \pi_r(\psi, \downarrow(\gamma, w)) \\ & \Leftrightarrow & \exists\gamma{:}AF\ \pi_r(\psi, \downarrow(\gamma, w))\end{array}$$

'$\xrightarrow{*}$' denotes the rewriting step and is an equivalence transformation. The composition of the relational translation, the addition of (7), the equivalence preserving elimination of the R-predicate, the equivalence preserving elimination of the equations and the deletion of (7) makes up the functional translation.

Definition 6 (Functional translation) First, define an auxiliary function π_f that takes two arguments: a modal formula and a 'world variable'. π_f is defined inductively by

$$\begin{array}{rcll}\pi_f(p, w) & = & p(w) & \text{if } p \text{ is a propositional variable} \\ \pi_f(\Box\psi, w) & = & \forall\gamma{:}AF\ \pi_f(\psi, \downarrow(\gamma, w)) & \\ \pi_f(\Diamond\psi, w) & = & \exists\gamma{:}AF\ \pi_f(\psi, \downarrow(\gamma, w)) & \end{array}$$

and for the propositional connectives π_f is a homomorphic function.

A *functional translation* for a modal formula A with formula variables $p_1, \dots, p_n$ is defined by

$$\Pi_f(A) \stackrel{\text{def}}{=} \forall p_1, \dots, p_n \forall w{:}W\ \pi_f(A, w).$$

A set Φ of Hilbert axioms is translated as:

$$\Pi_f(\Phi) \stackrel{\text{def}}{=} (6) \wedge \bigwedge_{A \in \Phi} \Pi_f(A).$$

For Hilbert rules Π_f is defined like the relational translation function. ◁

Take for example the McKinsey axiom (2). $\Pi_f(\Box\Diamond p \Rightarrow \Diamond\Box p)$ is given by

$$\forall p \forall w{:}W\ (\forall\gamma{:}AF\ \exists\delta{:}AF\ p(\downarrow(\delta, \downarrow(\gamma, w)))) \Rightarrow (\exists\gamma{:}AF\ \forall\delta{:}AF\ p(\downarrow(\delta, \downarrow(\gamma, w)))).$$

By combining the previous results and noting that, for any model $\Im^*$

$$\Im^* \models (6) \wedge \Pi_r^*(\Phi) \wedge \neg\Pi_r^*(\varphi) \quad \text{iff} \quad \Im^* \models \Pi_f(\Phi) \wedge \neg\Pi_f(\varphi),$$

we obtain a soundness and completeness result for the functional translation.

Theorem 7 (Soundness and completeness of the functional translation) A modal formula φ is a Φ-theorem iff $\Pi_f(\Phi) \Rightarrow \Pi_f(\varphi)$ is a predicate logic theorem. ◁

The functional translation generates nested $\downarrow$-terms as arguments to predicates. We can avoid these by exploiting Corollary 5. It allows us to add sort declarations and axioms Ψ' to $\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi)$ while preserving satisfiability or unsatisfiability, provided Ψ' holds in the functional extension $\Im'$ of a relational model $\Im$, or there is an extension $\Im''$ of $\Im'$ that satisfies Ψ'. Let AF^* be a new sort and let $\circ$ be a new binary function symbol. Let Ψ' consist of

$$AF \sqsubseteq AF^* \tag{8}$$
$$\circ : AF^* \times AF^* \to AF^* \tag{9}$$
$$\forall x{:}W\ \forall \gamma, \delta{:}AF^*\ \downarrow(\gamma \circ \delta, x) = \downarrow(\delta, \downarrow(\gamma, x)) \tag{10}$$
$$\circ \text{ is associative} \tag{11}$$

If we define $\Im''$ as an extension of $\Im'$ in which AF^* is interpreted as the set of all possible compositions of functions in $AF^{\Im'}$, and $\circ$ is interpreted as ordinary composition of functions, then, of course, $\Im''$ satisfies Ψ'. In the presence of condition (10) and the associativity property of $\circ$ we use a more economic notation. Instead of nested $\downarrow$-terms, like $\downarrow(\delta, \downarrow(\gamma, x))$, we use flat terms, like $\downarrow((\gamma \circ \delta), x)$. We can economize even more, by writing $\downarrow([\gamma_1 \ldots \gamma_n], w)$ instead of $\downarrow(\gamma_1 \circ \ldots \circ \gamma_n, w)$ or $\downarrow(\gamma_n, \downarrow(\gamma_{n-1}, \ldots \downarrow(\gamma_1, w) \ldots))$. This is the 'world-path syntax' introduced in Ohlbach (1988a). We will use this notation in the remainder of this paper.

In (1988a) Ohlbach exhibits a syntactic invariant for the functionally translated terms. Each variable occurs with the *same prefix* of other variables (this is known as prefix stability or unique path property (Auffray and Enjalbert 1992)). More precisely, if a variable γ occurs in a term $\downarrow([\alpha_1 \ldots \alpha_k \gamma \ldots], w)$ part of a functional formula ψ then all other occurrences of γ in ψ have the same prefix $[\alpha_1 \ldots \alpha_k]$. This property is exploited in the following lemma.

Lemma 8 Let ψ be a formula in the functional language. Without loss of generality we assume ψ is in prefix normal form, i.e. ψ consists of a quantifier prefix followed by a quantifier-free matrix. Let γ be a free variable in ψ that occurs with the unique prefix α, which may be an arbitrary string of AF-terms, that is, the occurrences of γ in ψ are of the form $\downarrow([\alpha\gamma \ldots], w)$. Let $\Im$ be a functional model and let $u \stackrel{\text{def}}{=} \Im(\downarrow(\alpha, w))$.

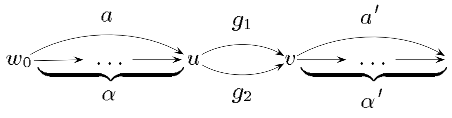
Then, if for two accessibility functions g_1 and g_2, $g_1(u) = g_2(u)$,

$$\Im[\gamma/g_1] \models \psi \quad \text{iff} \quad \Im[\gamma/g_2] \models \psi.$$

Proof Suppose the variables in the quantifier prefix α of ψ are $\beta_1, \ldots, \beta_n$ and ψ' is the matrix of ψ. The β_i may be existentially or universally quantified. Define two $\Im$-like interpretations:

$$\Im_1 \stackrel{\text{def}}{=} \Im[\gamma/g_1, \beta_1/b_1, \ldots, \beta_n/b_n] \quad \text{and} \quad \Im_2 \stackrel{\text{def}}{=} \Im[\gamma/g_2, \beta_1/b_1, \ldots, \beta_n/b_n].$$

By structural induction on ψ' we show that $\Im_1$ satisfies ψ' iff $\Im_2$ satisfies ψ'. Only the base case with $\psi' = p(\downarrow([\alpha\gamma\alpha'], w))$ is non-trivial. Since the prefix of γ is unique, γ does not occur in α and it does not occur in α'. Thus, the interpretation of α and α' does not depend on the assignment for γ. Let $\Im_1(\alpha) \stackrel{\text{def}}{=} a \stackrel{\text{def}}{=} \Im_2(\alpha)$, $\Im_1(\alpha') \stackrel{\text{def}}{=} a' \stackrel{\text{def}}{=} \Im_2(\alpha')$, $\Im_1(w) \stackrel{\text{def}}{=} w_0 \stackrel{\text{def}}{=} \Im_2(w)$ and $g_1(u) \stackrel{\text{def}}{=} v \stackrel{\text{def}}{=} g_2(u)$. We have the following situation



and get

$$\begin{aligned}
\Im_1(\downarrow(\alpha \circ \gamma \circ \alpha', w) &= (a \circ g_1 \circ a')(w_0) \\
&= a'(g_1(a(w_0))) \\
&= a'(g_1(u)) \\
&= a'(g_2(u)) \\
&= a'(g_2(a(w_0))) \\
&= (a \circ g_2 \circ a')(w_0) \\
&= \Im_2(\downarrow(\alpha \circ \gamma \circ \alpha', w)
\end{aligned}$$

It follows that $\Im_1(\downarrow([\alpha\gamma\alpha'], w)) \in \Im_1(p)$ iff $\Im_2(\downarrow([\alpha\gamma\alpha'], w)) \in \Im_2(p)$, since $\Im_1(p) = \Im_2(p) = \Im(p)$. This proves the base case. The induction step is a straightforward application of the induction hypothesis. We omit the details and conclude, $\Im_1$ satisfies ψ' iff $\Im_2$ does too. There are no conditions on the assignments to the β_i variables. Thus, if a β_i is universally quantified, we take all assignments of accessibility functions to β_i, and if β_i is existentially quantified, we choose an appropriate assignment. We conclude: $\Im[\gamma/g_1] \models \psi$ iff $\Im[\gamma/g_2] \models \psi$. $\triangleleft$

Now we are ready to prove the main result of the paper. It forms the basis for the quantifier exchange rule that allows us to move existential quantifiers to the front.

Theorem 9 (Quantifier exchange rule) Let $\Im$ be a relational model of a modal formula φ, and let $\Im'$ be a functional extension of $\Im$. For the functional translation of φ the following equivalence is true in $\Im'$.

$$\begin{array}{c} \exists\vec{p}\exists w\exists\vec{\alpha}\forall\vec{\gamma}\exists\delta\ A(\downarrow([\vec{\alpha}_1\gamma_1\vec{\alpha}_2\ldots\vec{\alpha}_k\gamma_k\vec{\alpha}_{k+1}\delta\ldots], w)) \\ \Leftrightarrow \\ \exists\vec{p}\exists w\exists\vec{\alpha}\exists\delta\forall\vec{\gamma}\ A(\downarrow([\vec{\alpha}_1\gamma_1\vec{\alpha}_2\ldots\vec{\alpha}_k\gamma_k\vec{\alpha}_{k+1}\delta\ldots], w)), \end{array} \tag{12}$$

provided all occurrences of δ in A are prefixed by the same term $\vec{\alpha}_1\gamma_1\vec{\alpha}_2\ldots\vec{\alpha}_k\gamma_k\vec{\alpha}_{k+1}$, where

(i) $\vec{p}$ is short for $p_1, \ldots, p_n$,
(ii) $\vec{\gamma}$ is short for $\gamma_1, \ldots, \gamma_k$,
(iii) $\vec{\alpha}_l$ is short for $[\alpha_{l,1}\ldots\alpha_{l,i_l}]$ (note, $\vec{\alpha}_l$ may be empty),
(iv) $\vec{\alpha}$ is short for $[\alpha_{1,1}, \ldots, \alpha_{1,i_1}, \ldots \alpha_{k+1,1}, \ldots, \alpha_{k+1,i_{k+1}}]$,
(v) A is assumed to be in prefix normal form and may contain existential and universal quantifiers.

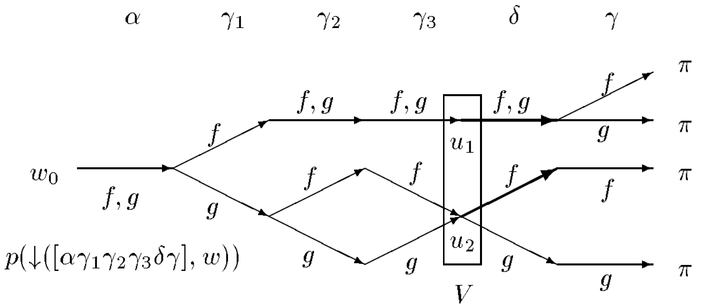
Proof The '$\Leftarrow$'-part is a valid predicate logic implication. For the '$\Rightarrow$'-part, suppose the left hand side of the equivalence is true in $\Im'$. Define a $\Im'$-like interpretation

$$\Im_1 \stackrel{\text{def}}{=} \Im'[\vec{p}/\vec{\pi}, w/w_0, \vec{\alpha}/\vec{a}] \quad \text{such that} \quad \Im_1 \text{ satisfies } \forall\vec{\gamma}\exists\delta\ A.$$

Let V be the set of pairs (u, D) such that

(i) $\Im_1' \stackrel{\text{def}}{=} \Im_1[\gamma_1/g_1, \ldots, \gamma_k/g_k, \delta/d] \models A$,
(ii) $\Im_1'([\downarrow(\vec{\alpha_1}\gamma_1\vec{\alpha_2}\ldots\vec{\alpha_k}\gamma_k\vec{\alpha_{k+1}}], w)) = u$ and
(iii) $d \in D$ and D is the maximal such subset of $AF^{\Im'}$.

Let $d_0 \in AF^{\Im'}$ be such that for all $(u, D) \in V$, $d_0(u) = d(u)$ for some $d \in D$. Such a d_0 exists because according to the definition of $\Im'$ (in Definition 4), $AF^{\Im'}$ is a maximal set of accessibility functions. The following picture



illustrates a typical model for the formula

$$\exists p \exists w \exists \alpha \forall \gamma_1, \gamma_2, \gamma_3 \exists \delta \forall \gamma \; p(\downarrow([a\gamma_1\gamma_2\gamma_3\delta\gamma], w)).$$

Here $V = \{(u_1, \{f, g\}), (u_2, \{f, g\})\}$. Choosing $d_0 = f$ (highlighted by the thick arrows) as assignment for δ makes this model satisfy the right hand side of (12).

In general, by the construction of V, we find for every $\Im_1$-like interpretation $\Im'_1$ defined by

$$\Im'_1 \stackrel{\text{def}}{=} \Im_1[\gamma_1/g_1, \ldots, \gamma_k/g_k, \delta/d] \models A$$

some pair $(u, D) \in V$ and some $d \in D$, where $u \stackrel{\text{def}}{=} \Im'_1([\downarrow(\vec{\alpha_1}\gamma_1\vec{\alpha_2}\ldots\vec{\alpha_k}\gamma_k\alpha_{\vec{k}+1}], w))$, $\Im'_1[\delta/d] \models A$ and $d(u) = d_0(u)$. Now, we can apply Lemma 8 and get

$$\Im'_1[\delta/d] \models A \quad \text{iff} \quad \Im'_1[\delta/d_0] \models A.$$

Because the assignments to the γ_i are arbitrary and because $\Im_1 \models \forall\vec{\gamma}\exists\delta\, A$ we can conclude that $\Im_1[\delta/d_0] \models \forall\vec{\gamma}\, A$. Hence $\Im_1 \models \exists\delta A$ and therefore, $\Im' \models \exists\vec{p}\exists w\exists\vec{\alpha}\exists\delta\forall\vec{\gamma}\, A$. $\lhd$

Since $A \Leftrightarrow B$ and $\neg A \Leftrightarrow \neg B$ are logically equivalent, the equivalence (12) remains true if both the left hand side and the right hand side are negated. This implies that in the functional translation of modal formulae, as well as in their negation, the quantifiers may be exchanged as we please, as long as the sort AF is interpreted (as in (ii) of Definition 4) as a maximal set of accessibility functions. In a first step we exploit this for moving existential quantifiers to the front in the negated form of the functional translation of a theorem.

Definition 10 (Quantifier exchange) Let Υ be the operation that converts a functionally translated modal formula into prenex normal form (consisting of a quantifier prefix followed by a quantifier free matrix) and moves all existential quantifiers of variables of sort AF inwards. $\lhd$

This means $\Upsilon(\Pi_f(A))$ has a quantifier prefix consisting of a sequence of universally quantified predicate variables followed by a sequence of universally quantified AF-variables and at the end of the prefix there is a sequence of existentially quantified

AF-variables. The negation $\neg\Upsilon(\Pi_f(A))$ of such a formula then has two sequences of existential quantifiers followed by a sequences of universal quantifiers.

Note, for any predicate logic formula A

$$A \Rightarrow \Upsilon(A) \quad \text{and} \quad \neg\Upsilon(A) \Rightarrow \neg A.$$

Corollary 11 (Quantifier exchange for modal theorems) Let Φ be a modal system extending KD. For any modal formula φ, φ is a Φ-theorem iff

$$\Pi_f(\Phi) \wedge \neg\Upsilon(\Pi_f(\varphi))$$

is inconsistent.

Proof By Theorems 2 and 7, φ is a Φ-theorem iff $\Pi_r(\Phi) \wedge \neg\Pi_r(\varphi)$ is inconsistent iff $\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi)$ is inconsistent. We prove

$$\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi) \text{ is consistent} \quad \text{iff} \quad \Pi_f(\Phi) \wedge \neg\Upsilon(\Pi_f(\varphi)) \text{ is consistent.}$$

If $\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi)$ is consistent, it has a functional model, and in this model, by Theorem 9, $\neg\Pi_f(\varphi)$ and $\neg\Upsilon(\Pi_f(\varphi))$ are equivalent. Hence, $\Pi_f(\Phi) \wedge \neg\Upsilon(\Pi_f(\varphi))$ is true in the model as well.

Now suppose $\Pi_f(\Phi) \wedge \neg\Upsilon(\Pi_f(\varphi))$ has a model. Υ moves existential quantifiers to the inside. Thus, $\Pi_f(\varphi)) \Rightarrow \Upsilon(\Pi_f(\varphi))$ and contrapositively $\neg\Upsilon(\Pi_f(\varphi)) \Rightarrow \neg\Pi_f(\varphi))$. Therefore, $\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi)$ has a model. $\lhd$

To illustrate the operation Υ of swapping quantifiers, consider McKinsey's axiom (2). Its functional translation $\Pi_f(M)$ is given by

$$\forall p\ \forall w{:}W\ (\forall\gamma{:}AF\ \exists\delta{:}AF\ p(\downarrow([\gamma\delta], w))) \Rightarrow (\exists\gamma'{:}AF\ \forall\delta'{:}AF\ p(\downarrow([\gamma\delta], w))).$$

The prefix normal form is

$$\forall p\ \forall w{:}W\ \exists\gamma{:}AF\ \forall\delta{:}AF\ \exists\gamma'{:}AF\ \forall\delta'{:}AF\ (p(\downarrow([\gamma\delta], w)) \Rightarrow p(\downarrow([\gamma\delta], w))).$$

Applying Υ yields

$$\forall p\ \forall w{:}W\ \forall\delta{:}AF\ \forall\delta'{:}AF\ \exists\gamma{:}AF\ \exists\gamma'{:}AF\ (p(\downarrow([\gamma\delta], w)) \Rightarrow p(\downarrow([\gamma\delta], w))).$$

The negation $\neg\Upsilon(\Pi_f(M))$ is now given by

$$\exists p\ \exists w{:}W\ \exists\delta{:}AF\ \exists\delta'{:}AF\ \forall\gamma{:}AF\ \forall\gamma'{:}AF\ (p(\downarrow([\gamma\delta], w)) \wedge \neg p(\downarrow([\gamma'\delta'], w))).$$

and in clause form with Skolem constants $a_\delta, b_{\delta'}$ and c_w:

$$\begin{array}{c} p(\downarrow([\gamma a_\delta], c_w)) \\ \neg p(\downarrow([\gamma' b_{\delta'}], c_w)). \end{array}$$

The Υ operation makes it possible to avoid Skolem functions completely in clause based refutational theorem proving. (In fact, for some decidable modal logics, this makes resolution a decision procedure.) For the *negated* translated theorems, Υ moves any existential quantifiers preceded by universal quantifiers outward so that existential quantifiers generate just Skolem constants, no complex Skolem terms. (This was already observed by A. Herzig.)

Applying Υ to a theorem φ results in a weaker formula φ', since $\varphi \Rightarrow \varphi'$, for $\exists x \forall y\ \psi(x,y) \Rightarrow \forall y \exists x\ \psi(x,y)$ is a theorem of predicate logic. Refuting $\neg\Upsilon(\varphi)$ instead of $\neg\varphi$ therefore means we are in fact proving a weaker version of the theorem than we originally wanted. Corollary 11 provides the conditions under which working with the weaker form does suffice for proving φ.

In this paper our main emphasis is not on translating *theorems* but our main emphasis is simplifying the translations of *Hilbert axioms.* As we argue in Section 2, one way of finding first-order formulae equivalent to the translation of any Hilbert axiom like the McKinsey axiom is by exchanging existential and universal quantifiers in the *negation* of the translation. Unfortunately, this weakens the axiom. This is certainly sound, any theorem that can be proved using weaker axioms also follows from the original axioms. However, weakening an axiom may be a source for incompleteness. We may not be able to prove all theorems in the weaker system.

There are two possibilities. First, we may try to exploit Theorem 9 which says that in the functional interpretations the quantifier exchange rule preserves equivalence. In the proof we needed the maximality condition of functional models. Unfortunately, this condition is not first-order definable. An infinite approximation is:

$$
\begin{aligned}
\mu \stackrel{\text{def}}{=} & (\forall x_1, x_2\ x_1 \neq x_2 \Rightarrow \\
& \quad \forall w\ \forall \gamma_1, \gamma_2\ \exists \delta\ {\downarrow}(\gamma_1, w) = {\downarrow}(\delta, w) \wedge {\downarrow}(\gamma_2, w) = {\downarrow}(\delta, w)) \wedge \\
& (\forall x_1, x_2, x_3\ x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \Rightarrow \\
& \quad \forall w\ \forall \gamma_1, \gamma_2, \gamma_3\ \exists \delta\ {\downarrow}(\gamma_1, w) = {\downarrow}(\delta, w) \wedge {\downarrow}(\gamma_2, w) = {\downarrow}(\delta, w)) \wedge {\downarrow}(\gamma_3, w) = {\downarrow}(\delta, w)) \\
& \vdots \quad \text{ad infinitum.}
\end{aligned}
$$

μ is true in our maximal functional models. By Corollary 5, we can add μ to $\Pi_f(\Phi) \wedge \neg\Pi_f(\varphi)$ without changing the consistency. Under the assumption μ, $\Pi_f(\Phi)$ and $\Upsilon(\Pi_f(\Phi))$ are equivalent. Under this assumption we are licensed to make use of quantifier exchange. But this may mean that we actually need to use μ in the process of finding a refutation. Since μ is infinite, this is not very practicable.

Another possibility for ensuring completeness is the following: If we prove instead of $\Pi_f(\varphi)$ the *weaker* theorem $\Upsilon(\Pi_f(\varphi))$, it may turn out that the weaker axioms are sufficient to prove the weaker theorems, without the assumption μ. The next corollary confirms that this is in fact possible.

Corollary 12 (Quantifier exchange for the axioms) For a modal system Φ extending KD, but with modus ponens and the necessitation rule as the only rules:

A modal formula φ is a Φ-theorem iff $\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(\varphi))$ is a theorem.

Proof *Soundness.* Suppose $\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(\varphi))$ is a theorem, Υ moves existential quantifiers *inside.* Thus, $\Pi_f(\Phi) \Rightarrow \Upsilon(\Pi_f(\Phi))$, and hence, $\Pi_f(\Phi) \Rightarrow \Upsilon(\Pi_f(\varphi))$ is a theorem. This implies $\Pi_f(\Phi) \wedge \neg\Upsilon(\Pi_f(\varphi))$ is inconsistent. Therefore, by Corollary 11, φ is a Φ-theorem.

Completeness. Suppose φ is a Φ-theorem. Then φ is either an instance of an axiom in Φ or it can be obtained by repeatedly applying the rules of the Hilbert system. The desired result follows by induction on the length of the proof sequence if we can show, (i) for all instances ψ' of axioms ψ in Φ,

$$\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(\psi')),$$

and (ii) for all applications of Hilbert rules 'from A_1 and ... and A_n derive A' in Φ,

$$\bigwedge_i (\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(A_i))) \Rightarrow (\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(A))).$$

(i) Let ψ be an axiom and let $\psi' \stackrel{\text{def}}{=} \psi[\vec{p}/\vec{\xi}]$ be an instance of ψ. We have

$$\Upsilon(\Pi_f(\Phi)) \Rightarrow \Upsilon(\Pi_f(\psi')) \text{ iff } \Upsilon(\Pi_f(\psi)) \Rightarrow \Upsilon(\Pi_f(\psi')) \text{ iff } \forall \vec{p}\ \psi_f \Rightarrow \psi_f[\vec{p}/\vec{\xi}],$$

which is true with $\forall \vec{p}\ \psi_f \stackrel{\text{def}}{=} \Upsilon(\Pi_f(\psi))$.

(ii) The proof for the rules modus ponens and necessitation is trivial. $\lhd$

If there are other rules in the Hilbert system, one must prove case (ii) for these rules individually. In general this should not be a problem.

This result allows us to pull in the negated functionally translated Hilbert axioms all existential quantifiers to the front. We may then apply the SCAN algorithm for finding an equivalent first-order formulation. Since in the clause form of the transformed Hilbert axioms no Skolem functions appear, only Skolem constants, it is always possible to reverse the Skolemization, provided the C-resolution step terminates.

Above we derived $\neg\Upsilon(\Pi_f(M))$ for the McKinsey axiom

$$\exists p\ \exists w{:}W\ \exists\delta{:}AF\ \exists\delta'{:}AF\ \forall\gamma{:}AF\ \forall\gamma'{:}AF\ (p(\downarrow([\gamma\delta],w)) \wedge \neg p(\downarrow([\gamma'\delta'],w)))$$

and its clause form:

$$p(\downarrow([\gamma a_\delta], c_w))$$
$$\neg p(\downarrow([\gamma b_{\delta'}], c_w)).$$

C-resolution yields:

$$\downarrow([\gamma a_\delta], c_w) \neq \downarrow([\gamma' b_{\delta'}], c_w).$$

Unskolemized:

$$\exists w\ \exists\delta,\delta'\ \forall\gamma,\gamma'\ \downarrow([\gamma\delta],w) \neq \downarrow([\gamma'\delta'],w)$$

and negated:

$$\forall w\ \forall\delta,\delta'\ \exists\gamma,\gamma'\ \downarrow([\gamma\delta],w) = \downarrow([\gamma'\delta'],w), \tag{13}$$

and this is first-order. Note that it is not neccessary to swap all existential quantifiers. In the case of the McKinsey axiom swapping one quantifier suffices. The resulting formula is

$$\forall w\ \forall\delta\exists\gamma\forall\delta'\exists\gamma'\ \downarrow([\gamma\delta],w) = \downarrow([\gamma'\delta'],w).$$

This is slightly stronger than (13), but it is still first-order.

In the Introduction we argued that the functional language seems more expressive than the relational language and therefore properties of the frame which are second-order in the relational language may become first-order in the functional language. Although, our argumentation is suggestive, from a purist's point of view, we did not prove that this is actually the case. We only showed that we can prove that a formula

φ is a Φ-theorem by proving a weaker theorem from weaker frame properties, which may be first-order, whereas the original frame properties are still second-order, even in the functional language. But from a practical and theorem proving point of view, we achieved the desired effect. We 'massaged' the second-order frame properties into first-order frame properties without changing the theorems.

5 Other modal systems

The functional translation has been defined for other modal systems: for non-serial systems, i.e. systems without the axiom D, for multi-modal systems and even for quantified modal systems. The results of the last section can be transferred to some of these systems.

Non-serial modal systems

Predicate logic as a target logic for the functional translation has no built-in facilities for dealing with partial functions. An 'accessibility function term' $\downarrow(\gamma, w)$ will always have an interpretation. It must denote something. In non-serial frames, however, a world w may not have an R-successor and therefore $\downarrow(\gamma, w)$ cannot denote a world accessible from w. This means the equivalence

$$\forall x, y\ R(x, y) \Leftrightarrow \exists\gamma{:}AF\ y = \downarrow(\gamma, w) \tag{14}$$

is not valid in non-serial frames. The standard solution is: encoding every partial function γ as a total function which maps the elements for which γ is not defined to elements of a special new sort $\perp$. $\perp$ stands for 'undefined' or just 'bottom'. Accordingly any formula in which such an 'undefined' situation may occur is translated into a conditional formula which handles the undefined cases.

In our context we introduce the sorts $\perp$ and $W^\perp$ with $W \sqsubseteq W^\perp$ and $\perp \sqsubseteq W^\perp$. The declaration for $\downarrow$ is now: $\downarrow$: $AF \times W^\perp \to W^\perp$. Furthermore, a new predicate symbol de (short for dead end) is introduced and instead of (14) we define R by

$$\forall x, y{:}W\ \ R(x, y) \Leftrightarrow \neg de(w) \Rightarrow \exists\gamma{:}AF\ y = \downarrow(\gamma, w) \tag{15}$$

We adapt the functional translation function π_f for modal formulae as follows:

$$\begin{array}{rcl}\pi_f(\Box\psi, w) & = & \neg de(w) \Rightarrow \forall\gamma{:}AF\ \pi_f(\psi, \downarrow(\gamma, w)) \\ \pi_f(\Diamond\psi, w) & = & \neg de(w) \wedge \exists\gamma{:}AF\ \pi_f(\psi, \downarrow(\gamma, w))\end{array}$$

Given a relational interpretation $\Im$ its functional extension $\Im'$ is required to satisfy the following:

(i) The predicate de is interpreted as the set of all end points of $R^\Im$, i.e.

$$de^{\Im'} \stackrel{\text{def}}{=} \{u \in W^{\Im'} \mid \neg\exists v\ R^\Im(u, v)\}.$$

(ii) $AF^{\Im'} \stackrel{\text{def}}{=} \{\gamma{:}W^{\perp\Im'} \to W^{\perp\Im'} \mid \text{if } \exists y\ R^\Im(x, y) \text{ then } R^\Im(x, \gamma(x)) \text{ else } \gamma(x) \in \perp^{\Im'}\}$.

With these modifications to Definition 4 all considerations for the serial case can be generalized and transferred to the non-serial case.

Multi-modal systems

Multi-modal logics have several different pairs of modal operators $\Box_i$ and $\Diamond_i$, each pair is associated with a separate accessibility relation R_i. The Hilbert axioms may specify individual properties of each of the R_i as well as interactions between different R_i.

The functional translation of multi-modal systems is a straightforward extension of the functional translation defined in Definition 6. For each accessibility relation R_i a sort AF_i is introduced and (14) is formulated for each AF_i (for R_i serial)

$$\forall x, y\ R_i(x, y) \Leftrightarrow \exists \gamma{:}AF_i\ y = \downarrow(\gamma, w).$$

It is easy to verify that the results for the mono-modal systems carry over to the multi-modal case.

Quantified modal systems

In the introduction we showed that in quantified modal logics the interpretation of a subformula of the formula $\Box(\exists x\ (p(x) \wedge \Box\Diamond\neg p(x)))$ in a particular world may depend on the path we follow to reach this world. Therefore, if there are domain quantifiers moving existential quantifiers to the front is not always possible. It remains to be investigated whether exchanging quantifiers is possible at least for some cases, for example, for the case that domain variables do not occur in different modal contexts. But before we start such an investigation, we need a proper reference system that can take over the role of the Hilbert systems in the propositional case. Recent incompleteness results found by Gasquet (1994) suggest that our proposals cannot be readily applied to Hilbert systems with quantifiers (Hughes and Cresswell 1984).

6 Summary

Thus far the functional translation of modal formulae has been investigated for the theorems to be proved in some modal logics with first-order frame properties. For the case of propositional modal logic it is possible to move in the functional translation of the negated theorem all existential quantifiers over universally quantified 'accessibility functions' to the front. The effect is that complex Skolem functions are avoided and at most Skolem constants occur in the clause normal forms. This was known.

We have shown, the same can be said for the functional translation of Hilbert axioms. We demonstrated that the McKinsey axiom which is not first-order in the relational context has corresponding first-order functional formulation. This formulation is computed automatically using quantifier elimination algorithms like SCAN, if in the negation of the translated axiom some existential quantifiers are moved to the front. This is not possible in the relational language. We have shown that swapping the existential and the universal quantifiers in the functional language preserves the theorems of the logic, provided the quantifiers are also swapped in the theorem which we wish to prove.

Using the methods proposed in this paper, we can now apply first-order predicate logic theorem proving techniques to a wider class of modal systems than was possible to this point.

References

Ackermann, W. (1935), Untersuchung über das Eliminationsproblem der mathematischen Logik, *Mathematische Annalen* **110**, 390–413.

Ackermann, W. (1954), *Solvable Cases of the Decision Problem*, North–Holland.

Auffray, Y. and Enjalbert, P. (1992), Modal theorem proving: An equational viewpoint, *Journal of Logic and Computation* **2**(3), 247–297.

Brachman, R. and Schmolze, J. (1985), An overview of the KL-ONE knowledge representation system, *Cognitive Science* **9**(2), 171–216.

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

Fariñas del Cerro, Luis and Herzig, A. (1988a), Linear modal deductions, *in* E. Lusk and R. Overbeek (eds), *Proc. of 9^{th} International Conference on Automated Deduction, LNCS 310*, Springer Verlag, pp. 487–499.

Fariñas del Cerro, Luis and Herzig, A. (1988b), Quantified modal logic and unification theory, *Rapport LSI 293*, Languages et Systèmes Informatique, Université Paul Sabatier, Toulouse.

Fattorosi-Barnaba, M. and de Caro, F. (1985), Graded modalities, *Studia Logica* **44**, 197–221.

Fine, K. (1969), *For some Propositions and so many Possible Worlds*, PhD thesis, University of Warwick.

Fine, K. (1972), In so many possible worlds, *Notre Dame Journal of Formal Logic* **13**(4), 516–520.

Gabbay, D. M. and Ohlbach, H. J. (1992), Quantifier elimination in second–order predicate logic, *in* B. Nebel, C. Rich and W. Swartout (eds), *Principles of Knowledge Representation and Reasoning (KR92)*, Morgan Kaufmann, pp. 425–435. Also published as a Technical Report MPI-I-92-231, Max-Planck-Institut für Informatik, Saarbrücken, and in the *South African Computer Journal*, 1992.

Gasquet, O. (1994), *Déduction automatique en logique multi-modale par traduction*, PhD thesis, Université Paul Sabatier, Toulouse, France.

Goble, L. F. (1970), Grades of modality, *Logique et Analyse* **13**, 323–334.

Herzig, A. (1989), *Raisonnement automatique en logique modale et algorithmes d'unification.*, PhD thesis, Université Paul-Sabatier, Toulouse.

Hughes, G. and Cresswell, M. (1984), *A Companion to Modal Logic*, Methuen & Co., London.

Jónsson, B. and Tarski, A. (1951), Boolean algebras with operators, Part I, *American Journal of Mathematics* **73**, 891–939.

Kripke, S. A. (1959), A completeness theorem in modal logic., *Journal of Symbolic Logic* **24**, 1–14.

Kripke, S. A. (1963), Semantical analysis of modal logic I, normal propositional calculi., *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **9**, 67–96.

Ohlbach, H. J. (1988a), A resolution calculus for modal logics, *in* E. Lusk and R. Overbeek (eds), *Proc. of* 9^{th} *International Conference on Automated Deduction, CADE-88 Argonne, IL*, Vol. 310 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, New York, pp. 500–516. Extended version appeared in (Ohlbach, 1988b).

Ohlbach, H. J. (1988b), A resolution calculus for modal logics, *SEKI Report SR-88-08*, FB Informatik, Universität Kaiserslautern, Germany. PhD Thesis, short version appeared in (Ohlbach, 1988a).

Ohlbach, H. J. (1994), Synthesizing semantics for extensions of propositional logic, *Technical Report MPI-I-94-225*, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Ohlbach, H. J. (1995), Boolean algebras with functions – correspondence, completeness and quantifier elimination, Submitted to IJCAI 95.

Ohlbach, H. J., Gabbay, D. and Plaisted, D. (1994), Killer transformations, *Technical Report MPI-I-94-226*, Max-Planck-Institut für Informatik, Saarbrücken, Germany. To be published in *Proc. of the 1993 Workshop on Proof Theory in Modal Logic*, Hamburg.

Sahlqvist, H. (1975), Completeness and correspondence in the first and second order semantics for modal logics, *in* S. Kanger (ed.), *Proceedings of the 3rd Scandinavian Logic Symposium, 1973*, North Holland, Amsterdam, pp. 110–143.

Schild, K. (1991), A correspondence theory for terminological logics: Preliminary report, *Proc. of IJCAI 91*, Morgan Kaufmann, pp. 466–471.

Schmidt-Schauß M. and Smolka, G. (1991), Attributive concept descriptions with complements, *Artificial Intelligence* **48**(1), 1–26.

Simmons, H. (1994), The monotonous elimination of predicate variables, *Journal of Logic and Computation*.

Szałas, A. (1992), On correspondence between modal and classical logic: Automated approach, *Technical Report MPI–I–92–209*, Max-Planck-Institut für Informatik, Saarbrücken.

van Benthem, J. (1984), Correspondence theory, *in* G. Dov M and F. Guenthner (eds), *Handbook of Philosophical Logic, Vol. II, Extensions of Classical Logic, Synthese Library Vol. 165*, D. Reidel Publishing Company, Dordrecht, pp. 167–248.

van der Hoek, W. (1992a), *Modalities for Reasoning about Knowledge and Quantities*, PhD thesis, Vrije Universiteit Utrecht.

van der Hoek, W. (1992b), On the semantics of graded modalities, *Journal of Applied Non–Classical Logics* **2**(1), 81–123.

Walther, C. (1987), *A Many-Sorted Calculus Based on Resolution and Paramodulation.*, Research Notes in Artificial Intelligence, Pitman Ltd., London.

Zamov, N. (1989), Modal resolutions, *Izvestiya VUZ. Mathematika* **33**(9), 22–29. Also published in *Soviet Mathematics*, Allerton Press.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the research group *Logic of Programming* at the Max-Planck-Institut für Informatik. They are available by anonymous ftp from our ftp server `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Regina Kraemer
Im Stadtwald
D-66123 Saarbrücken
GERMANY
e-mail: `kraemer@mpi-sb.mpg.de`

MPI-I-95-2-001	S. Vorobyov	Subject Reduction and Proof Normalization in Extensions of Fsub
MPI-I-94-261	P. Barth, A. Bockmayr	Finite Domain and Cutting Plane Techniques in CLP($\mathcal{PB}$)
MPI-I-94-257	S. Vorobyov	Structural Decidable Extensions of Bounded Quantification
MPI-I-94-252	P. Madden	A Survey of Program Transformation With Special Reference to *Unfold/Fold* Style Program Development
MPI-I-94-251	P. Graf	Substitution Tree Indexing
MPI-I-94-246	M. Hanus	On Extra Variables in (Equational) Logic Programming
MPI-I-94-241	J. Hopf	Genetic Algorithms within the Framework of Evolutionary Computation: Proceedings of the KI-94 Workshop
MPI-I-94-240	P. Madden	Recursive Program Optimization Through Inductive Synthesis Proof Transformation
MPI-I-94-239	P. Madden, I. Green	A General Technique for Automatically Optimizing Programs Through the Use of Proof Plans
MPI-I-94-238	P. Madden	Formal Methods for Automated Program Improvement
MPI-I-94-235	D. A. Plaisted	Ordered Semantic Hyper-Linking
MPI-I-94-234	S. Matthews, A. K. Simpson	Reflection using the derivability conditions
MPI-I-94-233	D. A. Plaisted	The Search Efficiency of Theorem Proving Strategies: An Analytical Comparison
MPI-I-94-232	D. A. Plaisted	An Abstract Program Generation Logic
MPI-I-94-230	H. J. Ohlbach	Temporal Logic: Proceedings of the ICTL Workshop
MPI-I-94-229	Y. Dimopoulos	Classical Methods in Nonmonotonic Reasoning
MPI-I-94-228	H. J. Ohlbach	Computer Support for the Development and Investigation of Logics
MPI-I-94-227	C. Weidenbach	Minimal Resolution
MPI-I-94-226	H. J. Ohlbach, D. Gabbay, D. Plaisted	Killer Transformations