Logic morphisms as a framework for backward transfer of lemmas and strategies in some modal and epistemic logics

Ricardo Caferra, Stéphane Demri, Michel Herment

LIFIA-IMAG,
46, Av. Félix Viallet,
38031 Grenoble Cedex, FRANCE
{caferra | stephane | herment}@lifia.imag.fr (uucp)

Abstract

There exist methods in automated theorem proving for non-classical logics based on translation of logics from a (non-classical) source logic (abbreviated henceforth SL) into a (classical) target logic (abbreviated henceforth TL). These valuable methods do not address the important practical problem of presenting proofs in SL. We propose a framework applicable at least to $S_4(p)$, K, T, $K_4$ for presenting proofs of theorems of these logics found in a familiar TL: Order-Sorted Predicate Logic (abbreviated henceforth OSPL). The method backward translates lemmas in a deduction (in TL) either (a) into lemmas in a corresponding deduction in SL (in the best case), or (b) into formulas semantically related to lemmas in a corresponding deduction (in the worst case). As a natural consequence we bring to the fore the fact that this framework can also be used to help in solving another important and very difficult problem: the transfer of strategies from one logic to another. One conjecture - with corresponding theorem which is a particular case of it - is stated. When (b) above holds we give sufficient (and in general satisfactory) conditions in order to obtain the lemmas in SL. Two examples are treated in full detail: the well known problem of the “wise man puzzle” and another one which shows how our method can be used to transfer strategies.

1. Introduction

Relating logics is a technique that has been used in automated deduction for non classical logics: . By E. Orlowska, who introduced the notion of resolution-interpretablity of a logic in another logic and applied it in order to construct theorem proving systems for algorithmic and m-valued Post logics (Orlowska 79, 80).
. More recently by A. Hertzig and H-J. Ohlbach who emphasize the idea of logic morphism which is implicitly used in the work of E. Orlowska (Herzig 89, Ohlbach 89).

The works of Ohlbach and Hertzig, in which unification plays a central role, were applied to several classes of modal and temporal logics. In these works the notions of source logic (SL) and target logic (TL) can be identified. SL is a logic to which we want to transfer results, in which we want to prove theorems and for which we do not have a good theorem prover or a theorem prover at all.... TL is a logic for which we know a lot of results, for which we have good theorem provers with good complete strategies...

In automated deduction for non-classical logics we can clearly, either use existing methods for these logics: tableau-based (see for ex. (Fitting 83)) or resolution-based (see for ex. (Enjalbert & Farías del Cerro 1989)), or use translation methods (i.e. based on logic morphism). Our work is set in the latter context.

A common feature of all works centered on logic morphisms is that they do not worry about a very important practical problem: presenting proofs in the source logic. In a fully general approach this amounts to translating proofs between (arbitrary) logics and, obviously, we shall not attack this problem. Instead, we shall deal with the problem of translating (for certain logics) some formulas of a proof found in the TL into formulas in a corresponding proof in the SL.

We shall identify conditions allowing backward translation of lemmas of proofs from a particular TL: the Order-Sorted Predicate Logic (OSPL) into some different (non classical) SL: $S_4(p)$, $K$, $T$, $K_4$. OSPL is a “good” TL because very good theorem provers with good complete strategies are available for it. In order to expound our work we shall limit ourselves in this paper to the multi-modal propositional logic $S_4(p)$ but it will distinctly appear that similar results can be straightforwardly obtained for K, T and K4.

A nice “by-product” of our results about transfer of lemmas is their usefulness as a framework for studying strategies transfer. It is well known that finding good strategies (and if it is possible to prove their completeness) is one of the central problems in automated deduction. If a strategy has been used to obtain a proof in TL, the steps of the proof can be considered as “keeping trace” of the strategy. Therefore a framework for studying the transfer of lemmas is also a framework for studying the transfer of strategies from TL to SL.

The structure of the paper is the following: in section 2 we present the basic definitions concerning logic
morphism, a recall of S4(p) (S4 with several agents) and the
definition of a partial inverse morphology. Section 3
contains the definition of a particular partial inverse
morphism between OSPL and S4(p). Section 4 contains
the main results of this paper: we state one conjecture and
prove one related theorem. A well known example ("the
wise man puzzle") is treated in detail. Another example
shows the possibilities of using our approach as a
framework for translating strategies. We have chosen the
latter example in order to compare our method with a
completely different one. Recently (Auffray, Enjalbert, &
Hebrard 1990) proved that the unit strategy is complete.

Recently (Auffray, Enjalbert, & Hebrard 1990) proved that the unit strategy is complete.

presentation of the examples in section 4 is done
with our proof-editor system (Caferra, Demri, & Hermen 91). Section 5 gives some ideas of future work.

2. Basic definitions
We use the notion of logic morphism from (Ohlbach 89)
and we add the notion of partial inverse
morphism corresponding to the backward lemma transfer from TL to
SL. For the sake of simplicity we shall consider, both in
TL and SL only formulas in clausal form. We shall not
formally define the notions of logics, specification
morphisms and logic morphisms here but the main ideas
of the approach in (Ohlbach 89) can be summarized as follows. This approach is based on the proviso that: there
exist i) a (good) theorem prover for a target logic and ii) a
standard translation from a context logic (i.e. an
intermediate logic, abbreviated henceforth CL) to a target
logic. The task of writing theorem provers for new logics
is therefore changed in the task of finding the way of
translating these new logics to CL.

"Translating" means here translating the syntax and the
semantics. Ohlbach uses "morphism" to name this
translation. We shall keep this name in this paper. In
some standard cases the semantics of SL is syntactically
captured by terms of the context logic via an
axiomatization of the properties of semantic structures.
Moreover, CL distinguishes in its syntax, the context
terms which capture the semantics of SL (for instance,
"[" denotes the application function, "o" the composition
function and "OC" the initial world) from the domain
terms. The first-order logic, CL, S4(p) and OSPL can be
redefined with this formalism. Our definition of a partial
inverse morphism fits in this theoretical framework. The
following definition uses the notations from (Ohlbach 89).

Definition 1
A partial inverse morphism \( \varphi \) for the logic morphism \( \Psi \)
between the two logics L1 and L2 is a mapping such that:

- For \( \sigma \in \Sigma_1 \) (set of signatures for L1), \( \varphi(\sigma) \) is a partial
function between \( \sigma_2 \) (\( \Sigma_2(\sigma) \)) and \( \varphi_1(\sigma) \) (\( \varphi_1 \) and \( \varphi_2 \)
map a signature to a set of formulas). In the sequel we
shall note \( \varphi(F) \) the application of \( \varphi(\sigma) \) to \( F \) (if it exists).\(^*\)

We illustrate the use of morphisms and partial inverse
morphisms between logics by taking S4(p) and SL and
OSPL as TL. The logic S4(p) is widely considered as the
basic model for epistemic logics. In a world with \( p \) agents
the intended interpretation of \( \Box_1 A \) is "the agent \( i \) knows
that \( A \) is true". The syntax and semantics for S4(p) and
its associated resolution calculus are those used in
(Enjalbert & Farinas del Cerro 89). In the sequel we shall
use a slight modification of the logic morphism defined
in (Ohlbach 89): each agent generates its own context
sorts. We associate to OSPL the resolution method
defined in (Schmidt-Schauss 88).

3. Morphism and partial inverse morphism
The morphism we shall use is a slight modification of the one
between S4(p) and CL defined in (Ohlbach 89): the
context sorts are indexed by the different agents. For
example the modal part of the formula morphism is:

\[
\Psi_F(\Box_l f) = \forall x 'W->\forall W' \Psi(F).
\]

Let \( \Sigma \) be a set of clauses in S4(p) and let \( \Sigma' \) be the
translation of those clauses into OSPL. Let \( \Phi' \) a clause
provable from \( \Sigma' \) and let \( \Phi \) be the possible inverse
translation of \( \Phi' \) back into S4(p). We will define the
conditions to backward translate \( \Phi' \) and we will state, as a
conjecture, that \( \Phi \) is always provable from \( \Sigma \). We can
now define the partial inverse morphism.

As proofs in OSPL are sequences of clauses, each lemma
is a clause. Therefore it suffices to define backward
translation for clauses.

To translate a clause, we capture the underlying
semantics of the transition from one world to another by
introducing a modal operator. The principle of the
backward translation consists in gathering the literals
which have some context in common. Each literal \( L \) has
the form \( sP(p(C)) \) where \( s \in \{1, -\} \) (as usual \( \Lambda \) denotes
the empty string), \( P \) is a predicate symbol not appearing
in the axioms, \( pw \) is a projection operator introduced by
the morphism, and \( C \) is a context term. First of all, we
algorithmically define auxiliary functions which deal with
syntactic transformations.

* generate-op(\( t \)) := if \( t \) is a variable of sort 'W-\rightarrow \forall W' then
  \( \Box_l \) else \( \Box_l \) \( % \) is a context term %
* local-context ( (\( t_1 \) ... \( t_n \)) := if \( n=1 \) then generate-op(\( t \))
  else generate-op(\( t \)). local-context( (\( t_2 \), ..., \( t_l \)) \( % \) ' is the
  concatenation operator, the \( t_i \) are context terms %
* \( \beta(\langle x_1 \circ \ldots \circ x_n, OC \rangle) \) m := if m = n then \( OC \) else
  \( \langle x_{m+1} \circ \ldots \circ x_n, OC \rangle \) \( % \) 0 \( \leq \) m \( \leq \) n %

This last function takes into account the fact that each
context term denoting a path-world is equivalent to \( OC \) or
to a term of the form \( \langle t_1 \circ \ldots \circ t_n, OC \rangle \).

* \( \varphi_1(L) :=
  \% L = sP(pw(C)) and C is a context term %
  if C = OC then P
  else generate-op(\( t \)) . \varphi_1(sP(pw(\beta(C,1))))

Definition of \( \varphi \)
Let \( \mathcal{C} \) be a clause \( L_1 V \ldots V L_n \). We define a set of classes
of literals \( \{C_i, 1 \leq i \leq k \} \) \( k \leq n \) which is a partition of
the set of literals of \( \mathcal{C} \). Moreover the classes have the
following properties:

422 GENERAL DEDUCTION SYSTEMS
For $1 \leq j \leq k$, there exist $\alpha_j$ such that
\[
\forall \ i \leq j \leq k, \ \exists \alpha_j \text{ such that }
\%
L_1 = s_1; P_1 (p(w(t_{L_1} o ... o t_{L_{1i}}, OC))) \text{ or }
\%
s_1; P_1 (p(w(OC))) \%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
Transformation S (applied to a clause C):

Step 1: To modify C by applying R1 and R2 as long as possible with a higher priority to the deepest subformulae of C.

Step 2: Apply D1 to C with the same restrictions. If the resulting formula is equal to C then go to the step 3) else go to the step 1).

Step 3: The result of S is C, and C has the following form C = C1 \& ... \& Cn where each Ci is a S4(p)-clauses. We note S(C) = C1 \& ... \& Cn or C \iff \{ C1, ..., Cn \}.

Transformation T (applied to \( \varphi(L) \)):

Apply R3 to \( \varphi(L) \) as long as possible with a higher priority for the deepest subformulae of \( \varphi(L) \). Return the resulting formula.

To prove the theorem 2 the Monotonicity of Entailment Lemma from (Abadi & Manna 86), can be used.

Theorem 2

To prove the theorem 2 the Monotonicity of Entailment Lemma (Abadi & Manna 86), can be used.

Theorem 2

(1) S terminates for every clause C and \( t \equiv S_4(p)C \Rightarrow S(C) \) and (2) T terminates for every backward lemma \( \varphi(L) \) and \( t \equiv S_4(p)\varphi(L) \iff T(\varphi(L)) \).

Corollary

If the conditions 1), 2) and 3) below hold

1) \( L \) is a lemma verifying the conditions of the Conjecture.
2) \( C \) is a clause which has been deduced from \{C1, ..., Cn\}.
3) There exists \( \alpha \) (1 \( \leq \alpha \leq k \)) such that every disjunct of \( f_\alpha \) is a disjunct of \( T(\varphi(L)) \), then \( C \equiv S_4(p) \varphi(L) \).

The backward translation we built between OSPL and S4(p), and its properties considering the modal resolution system for S4(p) (Enjalbert & Farîñas del Cerro 89) can be straightforwardly adapted to some other normal modal logics. By keeping the same backward translation between OSPL and K, K4 or T the logic morphism between these logics is mainly contained in the one defined for the multi-modal logic (Ohlbach 89).

We have considered 26 axioms from which we would like to deduce \( \varphi(C) \) (A7). We translate these formulations into S4(p).

The translation of A7 is (A7) \( \forall x \in C_1 : 'W_\rightarrow RTW'B, \forall x \in B_1 : 'W_\rightarrow RTW'C, \forall P_{B}[x \in C_1, x \in B_2, @A] \) where @A is a constant of sort \( 'W_\rightarrow RTW'A \).

A1 \( P_1 \odot P_2 \odot P_3 \odot P_4 \odot P_5 \odot P_6 \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} \in \{ A, B, C \} \) means that the wise i has a white spot.

The three men can see each other and they know this. Whenever one of them has a white or black spots he knows that his colleagues know this and he knows also that his colleagues knows this from each other:

A2 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i \in \{ A, B, C \} \) and \( i \neq j \).

A3 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} = \{ A, B, C \} \).

A4 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} = \{ A, B, C \} \).

C knows that B does not know that the colour of his spot and C knows that B knows that A does not know the colour of his spot.

A5 \( C \rightarrow \rightarrow B P_B \) A6 \( C \rightarrow \rightarrow B P_B A \) for \( i \in \{ A, B, C \} \).

Example 1 (“Wisc Man Puzzle”)

We illustrate our method with a famous example from Mc Carthy. Its traditional form is:

“A certain king wishes to determine which of his three wise men is the wisest. He arranges them in a circle so that he will put a white or black spot on each of their foreheads but at least one spot will be white. In fact all three spots are white. He then offers his favor to the one who first tells him the color of his spot. After a while, the wisest announces that his spot is white. How does he know?”

To axiomatize this puzzle in S4(p), we assume that the three wise men are A, B, C and C is the wisest. At least one of them has a white spot and everyone knows that everybody else knows that his colleagues know this:

A1 \( P_1 \odot P_2 \odot P_3 \odot P_4 \odot P_5 \odot P_6 \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} \in \{ A, B, C \} \) means that the wise i has a white spot.

The three men can see each other and they know this. Whenever one of them has a white or black spots he knows that his colleagues know this and he knows also that his colleagues knows this from each other:

A2 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i \in \{ A, B, C \} \) and \( i \neq j \).

A3 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} = \{ A, B, C \} \).

A4 \( P_i (\rightarrow P_i, \rightarrow P_i, \rightarrow P_i) \) for \( i, j, k \in \{ A, B, C \} \) and \( \{ i, j, k \} = \{ A, B, C \} \).

C knows that B does not know that the colour of his spot and C knows that B knows that A does not know the colour of his spot.

A5 \( C \rightarrow \rightarrow B P_B \) A6 \( C \rightarrow \rightarrow B P_B A \) for \( i \in \{ A, B, C \} \).

The translation of A7 is (A7) \( \forall x \in C_1 : 'W_\rightarrow RTW'B, \forall x \in B_1 : 'W_\rightarrow RTW'C, \forall P_{B}[x \in C_1, x \in B_2, @A] \) where @A is a constant of sort \( 'W_\rightarrow RTW'A \).

The translation of A7 is (A7) \( \forall x \in C_1 : 'W_\rightarrow RTW'C, \forall x \in B_1 : 'W_\rightarrow RTW'B, \forall P_{B}[x \in C_1, x \in B_2, @A] \) where @A is a constant of sort \( 'W_\rightarrow RTW'A \).

The translation of A7 is (A7) \( \forall x \in C_1 : 'W_\rightarrow RTW'C, \forall x \in B_1 : 'W_\rightarrow RTW'B, \forall P_{B}[x \in C_1, x \in B_2, @A] \) where @A is a constant of sort \( 'W_\rightarrow RTW'A \).
We first present the proof of \( \varphi(L) \) in S4(p) (Figure 2) and then we show how the conjecture holds and how the corollary has been a guide to build the S4(p)-proof.

We prove \( \varphi(GC_3) \Rightarrow \varphi(GC_3') \) holds. Since every disjunct of \( \Box C \mathcal{B} \Box A \mathcal{P} C \) is a disjunct of \( \varphi(GC_3') \) by the application of the corollary \( GC_3 \Rightarrow S4(p) \varphi(GC_3) \) holds. We get similar results for the other backward translated generated clauses.

In SL, in order to find a proof of \( A_7 \) we can consider the lemma \( GC_3' \) of the proof of TL. In the modal system we first generate clauses until some clauses (\( GC_3 \)) which verify the above conditions (a simple criterion is presented in the corollary) are deduced. The sequel of the proof will favour the use of these clauses. If we consider that every deduced clause is a lemma, we can “almost” build the proof in SL (in this particular case it is possible to build the whole proof).

Example 2

The following formula (from (Auffray, Enjalbert, & Hebrard 90)) is a S4-theorem for which the proof in OPSL verifies the hypothesis of the Conjecture.

We consider the following S4-clauses:

\[
(C_1): A \mathcal{B} \mathcal{B} (B \mathcal{N} C \mathcal{B} \mathcal{B} B) \\
(C_2): A \mathcal{A} (C \mathcal{B} \mathcal{B} B) \mathcal{B} \mathcal{B} B \\
(C_3): \mathcal{A} \mathcal{A} C \\
(C_4): \mathcal{B} \mathcal{B} B
\]

We translate these formulas into OSPL:

\[
(C_1'): A(pw(C)) \mathcal{B} B(\mathcal{B} C \mathcal{B} (C(x,y))) \mathcal{B} D(x) \mathcal{B} B(pw(OC)) \\
(C_2'): C(x, y) \mathcal{B} (C(x, y)) \mathcal{B} D(x) \mathcal{B} B(pw(OC)) \\
(C_3'): \mathcal{C} \mathcal{A} \mathcal{A} C \\
(C_4'): \mathcal{B} (B(pw(OC)) \mathcal{B} B(\mathcal{B} C \mathcal{B} (C(x,y)))) \mathcal{B} D(x)
\]

A refutation has been obtained using a linear-input strategy (i.e. a linear strategy in which one at least of the clauses used in a resolution step is an input clause). The refutation presented by our proof-editor looks like:
Using results of the preceding sections we get the proof in S4 presented in the Figure 4.

(Figure 4)

It is easy to verify that by backward translating the proof found in OSPL, the proof is essentially the same that the one in (Auffray, Enjalbert, & Hebrard 90): the S4-proof implicitly uses the input strategy but the linear strategy is only partially used in the global proof. Clearly we did not prove the completeness of the unit strategy but, from a practical point of view the result is the same. On the other hand our approach is much more general because it allows us to study different strategies (or heuristics) and different logics.

5. Conclusion and future work

We have presented a framework for the backward transfer of lemmas and strategies and we have shown how it works on two detailed examples. It is natural to ask about the possibilities and limits of our approach. We succinctly analyse now both of them. Theoretically, if SL admits a logic morphism to TL and if there exists a proof calculus in SL then our method could be applied. But as the proof calculus considered for TL is resolution, a first step to extend the class of source logics to which our method could be applied, consists in considering logics with a resolution proof calculus. Among the existing resolution proof calculi for non-standard logics we shall study the following ones:

- the first-order modal logic S5 (Farinas del Cerro 84)
- the propositional temporal linear logic (Cavalli & Farinas del Cerro 84)
- some epistemic logics (Konolige 86).

For the two latter logics, the problem of defining a logic morphism to OSPL and an inverse partial morphism is open. In principle there is no theoretical impossibility to answer to it. Obviously our approach could be applied with some other proof calculi and we are investigating in this way.

We are presently studying a backward morphism for linear temporal logic, for propositional S5 and for first-order modal logics. The main lines of future work are:

- To use the ideas presented in the second example in order to experimentally help in the study of the possibilities for transferring strategies.
- To try to prove the conjecture stated in section 4.
- To characterize theoretically the SL, TL and logic morphism adapted to the transfer of strategies and, simultaneously.

References


