Unsorted functional translations

Carlos Areces [2]

FaMAF, Universidad Nacional de Córdoba
Córdoba, Argentina

Daniel Gorín[3]

SKS – DFKI Bremen
Bremen, Germany

Abstract

In this article we first show how the functional and the optimized functional translation from modal logic to many-sorted first-order logic can be naturally extended to the hybrid language $\mathcal{H}(@, \downarrow)$. The translation is correct not only when reasoning over the class of all models, but for any first-order definable class. We then show that sorts can be safely removed (i.e., without affecting the satisfiability status of the formula) for frame classes that can be defined in the basic modal language, and show a counterexample for a frame class defined using nominals.

Keywords: Automated theorem proving, functional translation, sorts.

1 Introduction

The *functional translation* is a tool for automated modal reasoning that appeared independently and almost simultaneously in a number of publications in the late 1980's and early 1990's (see, e.g. [12,13,6,20,3,4]). This translation maps formulas from modal languages to first-order logic in a satisfaction preserving way, much like the *standard translation* does (see [5]). But the functional translation uses a semantic alternative to relational structures, and it has been argued and empirically demonstrated, that it produces formulas that can be much more compact and with a shallower term structure than

[1] This work was partially supported by grant PICT XXXYYY

[2] Email: carlos.areces@gmail.com

[3] Email: daniel.gorin@dfki.de

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.com/locate/entcs

those obtained with the standard translation [15,9]. The two properties are crucial when attenpting to use first-order automated reasoning. Moreover, and unlike other satisfiability preserving translations tailored for automated reasoning (e.g., the layered translation of [1]), the functional translation can be used for reasoning over a wide range of model clases, beside the class of all models. The first result we discuss in this article is that the functional translation extends naturally to cover the hybrid language $\mathcal{H}(@,\downarrow)$ [2].

The functional translation is often introduced using a many-sorted first-order language to simplify presentation. This means, in practice, that we need to either use a theorem prover that handles many-sorted first-order logic (e.g., SPASS [19]) or simulate sorts in unsorted first-order logic introducing additional one-place predicate symbols. Both alternatives might have an impact on performance when we attempt to carry out automated theorem proving.

It is argued in [17], for example, that the simulation of sorts by way of proposition symbols leads to irrelevant inferences. A system like SPASS, avoids these inferences but, on the other hand, the additional complexity of the machinery needed to handle sorted inferences (in the case of SPASS, *well-sorted unification* [18]) needs to be accounted for. It is difficult to properly measure whether sorts help or hinder automated theorem provers, and we will not pursue this matter here. Instead, we will show that in certain cases one can simply safely "erase" all sort annotations without changing the satisfiability status of the formula. That this can be done in the case of the basic modal logic (when reasoning over the class of all models) was already observed by Hustadt and Schmidt in [10], albeit without proof.

In Sections 2 and 3 we introduce the basic and the *optimized* functional translations. To make the article self contained, we include the original proofs of soundness, with minor corrections and adaptations to accommodate nominals and other hybrid machinery. Based on these proofs, we develop in Section 4 our main result: when reasoning over any modally definable class of models it is safe to erase sort annotations from formulas obtained using the (optimized) functional translation. In Section 5, on the other hand, we prove that sort erasure is not sound when reasoning over a class that is definable by a pure, hybrid axiom, by providing a concrete counterexample.

2 Functional models, functional translation

Through all the article, we will work in the multi-modal hybrid language $\mathcal{H}(@,\downarrow)$. For a fixed signature consisting of a set of proposition symbols Prop, a set of nominals Nom and a set of relation symbols Rel, all pairwise disjoint, its formulas are given by:

$$\varphi ::= p \mid i \mid \neg\varphi \mid \varphi \wedge \varphi \mid @_i\varphi \mid {\downarrow} i.\varphi \mid [r]\varphi,$$

where $p \in \mathsf{Prop}$, $i \in \mathsf{Nom}$ and $r \in \mathsf{Rel}$. We shall freely employ the typical derived operators $\vee$, $\rightarrow$, $\langle r \rangle$, etc. with their usual meaning.

For the semantics we take as models pairs $\langle \mathcal{I}, g \rangle$, where $\mathcal{I} = \langle W, \cdot^{\mathcal{I}} \rangle$ is a relational interpretation such that $p^{\mathcal{I}} \subseteq W$ for $p \in \mathsf{Prop}$, and $r^{\mathcal{I}} \subseteq W \times W$, for $r \in \mathsf{Rel}$; while $g : \mathsf{Nom} \rightarrow W$ is an assignment for nominals. We can, then, give meaning to formulas of $\mathcal{H}(@, \downarrow)$ via the standard translation to first-order logic. That is, for $w \in W$ and $j \in \mathsf{Nom}$ not occurring in φ, we have:

$$\mathcal{I}, g, w \models \varphi \iff \mathcal{I} \models_{\text{FO}} ST_j(\varphi)[g^j_w]. \tag{1}$$

The standard translation commutes with boolean connectives and satisfies:

$$\begin{array}{rlrl} ST_j(p) & \overset{def}{=} p(j) & ST_j(i) & \overset{def}{=} i = j \\ ST_j(@_i\varphi) & \overset{def}{=} ST_i(\varphi) & ST_j(\downarrow i.\varphi) & \overset{def}{=} \exists i.(i = j \wedge ST_j(\varphi)) \\ ST_j([r]\varphi) & \overset{def}{=} \forall k.(r(j,k) \rightarrow ST_k(\varphi)) & (k \text{ is fresh}) & \end{array}$$

For a class of models $\mathcal{C}$, φ is $\mathcal{C}$*-satisfiable* ($\mathcal{C}$*-valid*, notation $\mathcal{C} \models \varphi$) if for some model (for every model) $\langle \mathcal{I}, g \rangle$ in $\mathcal{C}$, $\mathcal{I}, g, w \models \varphi$ for some w (for all w). If $\mathcal{C}$ is the class of all models, we say that φ is *satisfiable* (*valid*, notation $\models \varphi$). The underlying *frame* of a model $\langle \mathcal{I}, g \rangle$ is the restriction of $\mathcal{I}$ to symbols in Rel (i.e., ignoring Prop). As it is standard in modal logic, we are usually interested in classes defined as those models whose underlying frame satisfy certain condition (e.g., transitivity). Any such class $\mathcal{C}$ is said to be defined by a formula φ whenever $\langle \mathcal{I}, g \rangle$ is in $\mathcal{C}$ iff $\mathcal{I}, g, w \models \varphi$ for all w. See [5] for details.

We can regard the standard translation as a direct encoding in first-order logic of the semantic clauses for the modal operators. The translation is very simple but it is, in general, not suitable for translation-based automated reasoning. It is easy to find simple modal formulas which, when translated into first-order logic using ST and then solved via resolution, result in infinite clause sets (see [1]). This is the main motivation for the, arguably more complex, functional translation we will describe below.

The key to understand the functional translation is an alternative representation of relational structures. Assume for a moment that $\mathsf{Rel} = \{r\}$. Consider, then, Figure 1a which shows a relational structure with a domain consisting of three elements. One can alternatively represent this particular structure using three total functions f, g and h, and a predicate de, as long as the following property holds:

$$\forall xy.\big(r(x,y) \leftrightarrow (\neg de(x) \wedge (f(x) = y \vee g(x) = y \vee h(x) = y))\big). \tag{2}$$

We use de (for "dead end") to "mark" those states that have no r-successor, and f, g and h on each state to "witness" each r-successor. There are many

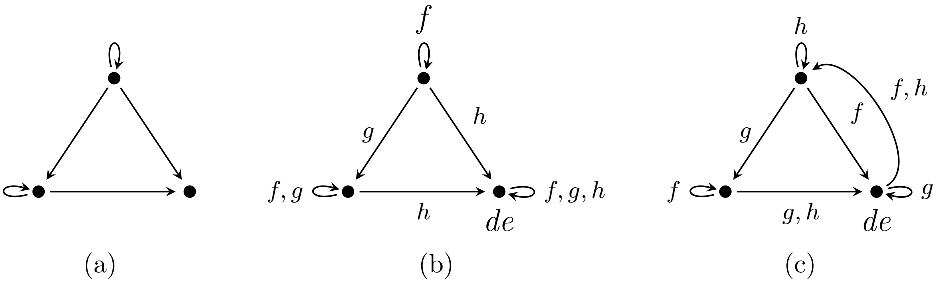


Fig. 1. Relational model (a) is expressed in (b) and (c) with functions f, g, h and a predicate de.

valid arrangements for f, g and h; Figures 1b and 1c show two such representations. It is straightforward to verify they both satisfy condition (2).

Proposition 2.1 *Let $\mathcal{I}$ be a finite first-order interpretation for a signature with a two-place relation symbol r. Then there exists an interpretation $\mathcal{I}'$ extending $\mathcal{I}$ to a signature that additionally contains a one-place relation symbol de and unary function symbols $f_1, f_2, \ldots, f_n$, such that:*

$$\mathcal{I}' \models_{\text{FO}} \forall xy. \big(r(x,y) \leftrightarrow (\neg de(x) \wedge (f_1(x) = y \vee f_2(x) = y \ldots \vee f_n(x) = y))\big).$$

Functions $f_1, f_2, \ldots, f_n$ can be alternatively represented using only one binary function f, that takes a function index as an additional argument. This is easily expressed in a language with two sorts ω and ι, the former will refer to proper nodes of the model, the latter to function indices.

Proposition 2.2 *Let $\mathcal{I}$ be a sorted first-order interpretation for a signature with a relation symbol of sort $r : \omega \times \omega$. Then there exists an interpretation $\mathcal{I}'$ extending $\mathcal{I}$ to a signature that additionally contains a relation symbol $de : \omega$ and a function symbol $f : \iota \times \omega \to \omega$, such that:*

$$\forall xy{:}\omega. \big(r(x,y) \leftrightarrow (\neg de(x) \wedge \exists z{:}\iota. f(z,x) = y)\big).$$

Unlike Proposition 2.1, this encoding is suitable for infinite interpretations.

We now have everything in place to define a notion of functional model. First, to each choice of Prop and Rel we assign a *functional (sorted) correspondence language* with sorts ω and ι, where each $p \in \mathsf{Prop}$ and each de_r ($r \in \mathsf{Rel}$) is a one-place predicate symbol of sort ω, and where there is a binary function symbol $f_r : \iota \times \omega \to \omega$ for each $r \in \mathsf{Rel}$.

Definition 2.3 [Functional models] A functional model is a many-sorted interpretation for the functional correspondence language, i.e., a structure $\mathcal{I} = \langle W, I, \cdot^{\mathcal{I}} \rangle$ where W and I are non-empty domains for sorts ω and ι, respectively; $p^{\mathcal{I}} \subseteq W$ for every $p \in \mathsf{Prop}$; and, for each $r \in \mathsf{Rel}$, $f_r : I \times W \to W$ and $de_r \subseteq W$.

Clearly, every functional model induces a relational model such that for every relation r the following holds:

$$\forall x, y{:}\omega . r(x, y) \leftrightarrow (\neg de_r(x) \wedge \exists z{:}\iota . f(z, x) = y). \qquad (3)$$

Therefore, we say that a functional model satisfies a modal formula φ if and only if its induced relational model satisfies φ. In Section 3 we will be interested in *maximal models*, that is, functional models where every possible function is realized.

Definition 2.4 [Maximal models] Consider a functional model $\mathcal{I} = \langle W, I, \cdot^{\mathcal{I}} \rangle$ and let $r^{\mathcal{I}}$ be the relation induced by (3) for each $r \in \mathsf{Rel}$. We say $\mathcal{I}$ is *maximal* if for each total function $\gamma : W \to W$ such that $(w, \gamma(w)) \in r^{\mathcal{I}}$ for all $w \in W$, there exists an $i \in I$ for which $f_r(i, x) = \gamma(x)$, for all x.

Any functional model can be extended to a maximal one without changing the induced relational model. Hence:

Proposition 2.5 *A formula of the basic modal logic is satisfiable iff there exists a (maximal) functional Kripke model that satisfies it.*

Putting all the pieces toghether, then, we can see the *functional translation* simply as a "standard" translation to many-sorted first-order logic over the functional correspondence language.

Definition 2.6 [FT] Let j be variable of sort ω occurring in a term t of the functional language. The functional translation to first-order logic FT_t maps formulas of $\mathcal{H}(@, \downarrow)$ into first-order logic formulas in the functional correspondence language with a free j as follows (it commutes with boolean connectives):

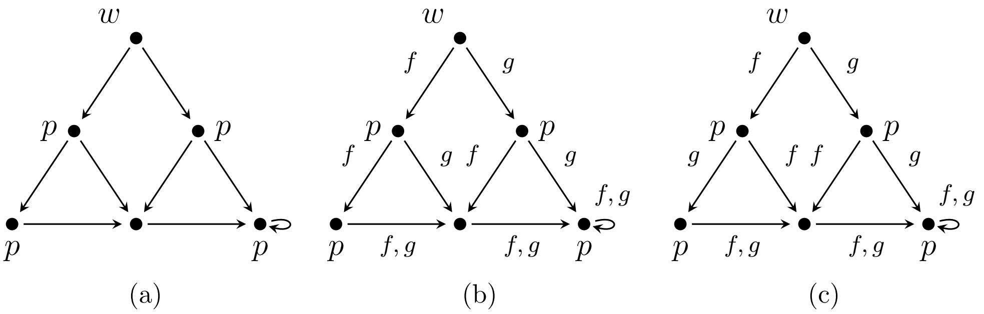
$$\begin{array}{rlrl}
FT_t(p) & \overset{def}{=} p(t) & FT_t(i) & \overset{def}{=} i = t \\
FT_t(\downarrow i.\varphi) & \overset{def}{=} \exists i{:}\omega.(i = t \wedge FT_t(\varphi)) & FT_t(@_i\varphi) & \overset{def}{=} FT_i(\varphi) \\
FT_t([r]\varphi) & \overset{def}{=} \neg de_r(t) \to \forall z{:}\iota . FT_{f_r(z,t)}(\varphi) & (z \text{ is fresh}) &
\end{array}$$

Theorem 2.7 *Let φ be a formula of $\mathcal{H}(@, \downarrow)$ and let i be a nominal not occurring in φ. Then the following hold:*

(i) $\models \varphi$ *iff* $\models_{\mathrm{FO}} \forall i{:}\omega . FT_i(\varphi)$.

(ii) φ *is satisfiable iff* $\exists i{:}\omega . FT_i(\varphi)$ *is satisfiable.*

It is straightforward to see that Theorem 2.7 can be extended to the case where we are reasoning with respect to a first-order definable class of models. For example, suppose we require r to be interpreted as a transitive relation, which is expressible in first-order logic as $\forall xyz.(r(x, y) \wedge r(y, z) \to r(x, z))$.



(a) (b) (c)

Fig. 2. A model (a) for formula (5) and two models (b) and (c) for its functional translation

By combining it with the equivalence (3) and performing some valid transformations we obtain the functional equivalent:

$$\forall x{:}\omega.(\neg de(x) \to \forall ab{:}\iota\exists c{:}\iota.f_r(b, f_r(a, x)) = f_r(c, x)). \tag{4}$$

The formula φ in $\mathcal{H}(@, \downarrow)$ will be satisfiable in the class of models where r is transitive if and only if the conjunction of (4) and $FT(\varphi)$ is satisfiable.

3 Optimized functional translations

Consider the following simple modal formula:

$$[r](p \to \langle r\rangle p). \tag{5}$$

By Theorem 2.7, this formula is satisfiable if and only if its functional translation is satisfiable too:

$$\exists i{:}\omega.(\neg de(i) \to \forall y{:}\iota.(p(f(y, i)) \to (\neg de(f(y, i)) \wedge \exists z{:}\iota.p(f(z, f(y, i)))))). \tag{6}$$

By skolemizing i and z we obtain the equisatisfiable formula:

$$\neg de(c) \to \forall y{:}\iota.(p(f(y, c)) \to (\neg de(f(y, c)) \wedge p(f(g(y), f(y, c))))). \tag{7}$$

This formula contains two skolem symbols: a constant c and a unary function g. The so-called "optimized functional translation" [14] guarantees that only *constants* need to be introduced during skolemization. Because skolem functions may cause complex terms to be built up during resolution, the optimized translation may drastically reduce the saturation process. Moreover, this simplifies the development of terminating resolution strategies [16].

To illustrate the idea behind the optimized translation, let us consider again formula (5). Figure 2a shows a model that satisfies (5) at node w. Figure 2b, on the other hand, shows a functional model for (7). It is easy to verify using (3) that this model induces the one of Figure 2a.

Observe now that if i is interpreted as w (notation: $i \mapsto w$) there are two possible values for y, namely f and g. If $y \mapsto f$, then we must pick $z \mapsto f$, while for $y \mapsto g$ we must select $z \mapsto g$. Therefore, the right value for z is effectively a function of y, as witnessed by the skolemization. But here comes the interesting part: we can "rearrange" the assignment of functions in a way that makes the choice of z independent of y. An example is shown in Figure 2c; this model also induces (a) but here the right choice is $z \mapsto g$ independent of the value of y. In maximal models (cf. Definition 2.4) where all possible "rearrangement" of functions are included, it is always possible to make the interpretation of each variable independent of the others.

Ohlbach and Schmidt [14] take advantage of this observation and prove that it is sound, in terms of satisfiability, to swap two consecutive quantifiers. Therefore one can take a formula obtained using the basic functional translation and simply make all the existential quantifiers come before universal ones, effectively avoiding the introduction of skolem functions. This is exactly what the optimized functional translation does.

Definition 3.1 The optimized functional translation to first-order logic OFT is defined as $OFT_j(\varphi) = \vartheta(FT_j(\varphi))$, where $\vartheta(\gamma)$ takes γ to prenex normal form and moves all existential quantifiers of sort ι to the front.

The soundness of this translation follows from the next proposition [14].

Proposition 3.2 *If γ is a formula in prenex normal form with a quantifier-free matrix δ and γ is equivalent to the functional translation of a modal formula, then γ is satisfiable iff $\exists\overline{x}{:}\iota\forall\overline{y}{:}\iota\delta$ is satisfiable too, where all the $\overline{x}$ and $\overline{y}$ are existentially and universally quantified, respectively, in γ.*

We will follow the proof given in [14] to verify that the result also holds in the hybrid case. The proof uses a syntactic invariant which functionally translated terms possess, known as "prefix stability" [12] or "unique path property" [4]. Intuitively, what this property says is that we can build a tree (or a forest) out of the set of terms and subterms occurring in a formula such that: i) nodes of the tree are terms, ii) arcs are labeled with variables of sort ι, iii) t_1 is the father of t_2 using an arc labeled by y iff $t_2 = f_r(y, t_1)$ for some $r \in \mathsf{Rel}$, and iv) every variable of sort ι labels only one arc.

Definition 3.3 [Prefix stability] We say a formula γ is *prefix-stable* if, given the set T_γ of all the terms occurring in γ, it holds that for every variable y of sort ι in T_φ, there exist a term t and a function symbol f such that if y occurs in a term in T_φ, then every occurrence of y is of the form $f(y,t)$. We will call $f(y,t)$ the *context of y in φ*.

As an example, consider the variable y that occurs in the functional translation of (5): all its occurrences have the same context, namely, $f(y,x)$. It is straightforward to see that this property follows from the way functional terms are built in the translation.

This syntactic property has a semantic counterpart. Suppose, again, that all occurrences of a variable y of sort ι are in the context $f(y,t)$ for fixed t and f. Then, for any functional model, the function "indexed" by y will be *relevant* only to determine successors of (the interpretation of) t. This is formally expressed in the following lemma.

Lemma 3.4 *Let γ be a prefix-stable formula in the functional correspondence language and let y be a free variable in γ that occurs in context $f_r(y,t)$, with all the variables in t free in γ. Furthermore, let $\mathcal{I} = \langle W, I, \cdot^{\mathcal{I}}\rangle$ be a functional model, g a (sorted) assignment and $a, b \in I$ such that $f_r^{\mathcal{I}}(a, g(t)) = f_r^{\mathcal{I}}(b, g(t))$, where $g(t)$ is the interpretation of term t using $\mathcal{I}$ and g. Then we have:*

$$\mathcal{I} \models_{\text{FO}} \gamma[g^y_a] \iff \mathcal{I} \models_{\text{FO}} \gamma[g^y_b]$$

Proof. The proof is by induction on γ. We look only at the base case. Assume γ is of the form $p(t')$, with $f_r(y,t)$ a subterm of t'. Let us define $g_a = g^y_a$ and $g_b = g^y_b$. The first thing to observe is that because y does not occur in t, we have $g_a(t) = g_b(t) = g(t)$. Therefore, we also have $g_a(f_r(y,t)) = g_b(f_r(y,t))$. Finally, again because of prefix-stability, we know there is no other occurrence of y in t' and, therefore, $g_a(t') = g_b(t')$, from which the expected result follows. An analogous reasoning can be used to handle the case where φ is an equality of the form $t_1 = t_2$. The inductive cases follow simply by inductive hypothesis.□

This lemma is a corrected version of *Lemma 4.6* in [14]. Indeed, in [14] variables in t are explicitly allowed to occur bound in γ, but in that case whether $\mathcal{I} \models_{\text{FO}} \gamma[g^y_a]$ holds need not depend on the value of $f_r^{\mathcal{I}}(a, g(t))$.[4] In any case, using Lemma 3.4 one can prove the following result (cf. [14, *Theorem 4.7*]).

Theorem 3.5 *Let γ be a prefix-stable formula in the functional correspondence language and let y be a free variable in γ that occurs in context $f_r(y,t)$, with all the variables in t free in φ. Finally, let $\mathcal{I} = \langle W, I, \cdot^{\mathcal{I}}\rangle$ be a* maximal *functional model. Then, for every assignment g we have:*

$$\mathcal{I} \models_{\text{FO}} \forall x_1 \ldots x_k{:}\iota \exists y{:}\iota.\gamma[g] \iff \mathcal{I} \models_{\text{FO}} \exists y{:}\iota \forall x_1 \ldots x_k{:}\iota.\gamma[g].$$

Proof. The right-to-left implication is already valid in the general case, so we only have to consider the left-to-right one. Suppose, then, that the antecedent holds. This means there must exist some function $\alpha : I^k \to I$ such that, $\mathcal{I} \models_{\text{FO}} \gamma[g^{x_1}_{a_1} \cdots {}^{x_k}_{a_k}{}^{y}_{\alpha(a_1 \ldots a_k)}]$ holds, for every $a_1 \ldots a_k \in I$. Now, let $b \in I$ be such that, $f_r^{\mathcal{I}}(b, g'(t)) = \alpha(a_1 \ldots a_k)$ for $g' = g^{x_1}_{a_1} \cdots {}^{x_k}_{a_k}$. Such a b must exist

[4] For a counterexample, take $\mathcal{I} = \langle \{u,v\}, \{a,b\}, \cdot^{\mathcal{I}}\rangle$, $f_r^{\mathcal{I}}(u,a) = f_r^{\mathcal{I}}(v,a) = f_r^{\mathcal{I}}(u,b) = v$, $f_r^{\mathcal{I}}(v,b) = u$, $p^{\mathcal{I}} = \{v\}$; $g(w) = v$, $g(z) = b$ and $\gamma = \forall z{:}\iota.p(f_r(y, f_r(z,w)))$, with $t = f_r(z,w)$.

since $\mathcal{I}$ is a maximal model. Therefore, using Lemma 3.4 we may conclude that $\mathcal{I} \models_{\text{FO}} \gamma[g'^{y}_{b}]$ must hold. But since b is independent of $a_1 \ldots a_k$, we finally obtain that $\mathcal{I} \models_{\text{FO}} \exists y{:}\iota \forall x_1 \ldots x_k{:}\iota.\gamma[g]$. □

Theorem 3.6 *Let φ be an $\mathcal{H}(@, \downarrow)$-formula and let i be a nominal not occurring in φ. Then the following hold:*

(i) $\models \varphi$ *iff* $\models_{\text{FO}} \forall i{:}\omega.OFT_j(\varphi)$.

(ii) φ *is satisfiable iff* $\exists i{:}\omega.OFT_i(\varphi)$ *is satisfiable.*

Proof. It is enough to prove that $FT_i(\varphi)$ is satisfiable iff $\vartheta(FT_i(\varphi))$. The right-to-left implication is valid in general. For the, other direction, suppose then that $\mathcal{I} \models_{\text{FO}} FT_i(\varphi)[g]$ for some $\mathcal{I}$ and g. Without loss of generality, we may assume that $\mathcal{I}$ is maximal. Let ψ be the result of taking $OFT_i(\varphi)$ and moving all existential quantifiers of sort ι after every universal quantifier. Observe that $FT_i(\varphi) \to \psi'$ is universally valid, and, therefore, $\mathcal{I} \models_{\text{FO}} \psi[g]$. Now, using Theorem 3.5, we can move every existential quantifier in ψ to the front, one at a time (for there must always exist one such that its bound variable y occurs in a context $f_r(y, t)$ and all the variables in t are either universally quantified or their existential quantifiers have been moved to the top already). This process can be repeated only finitely many times and the resulting formula ψ' satisfies $\mathcal{I} \models_{\text{FO}} \psi'[g]$ and is equivalent to $\vartheta(FT_i(\varphi))$. □

The above proof only requires that a maximal model always exists for a satisfiable formula. This means that the optimized functional translation also works when we are interested in satisfiability with respect to a first-order definable frame condition.

4 Sort erasure on modally definable classes of models

We are ready to discuss the soundness of the erasing sort annotations from the functional translations. Intuitively, what we need to see is that every satisfiable, functionally translated formula is satisfied by a functional model $\langle W, I, \cdot \rangle$ where the cardinalities of W and I match. Only maximal models pose a problem here, but we will see that closure under disjoint unions guarantees that one can raise the cardinality of W when needed.

Let us start by properly formalizing what we mean by *sort erasure.*

Definition 4.1 The sort erasure transformation $(\cdot)^-$ takes a many-sorted first-order formula and eliminates all sorts as follows:

$$\begin{aligned} a^- &= a, \text{ for } a \text{ a first-order atom} & (\varphi \wedge \psi)^- &= \varphi^- \wedge \psi^- \\ (\neg\varphi)^- &= \neg(\varphi^-) & (\exists x{:}\alpha.\varphi)^- &= \exists x.(\varphi)^-. \end{aligned}$$

An *unsorted functional model* is a model for the resulting signature.

Clearly, φ is not equivalent to φ^- in the general case. But consider again the model of Figure 1a. In Figures 1b and 1c it is "represented" (cf. Section 2) using three functions, but we can certainly represent it with any larger number of functions, since we do not care about duplicated functions.

On the other hand, since one of the nodes of this model has a fan-out of three, it cannot be represented with less than three functions. Because the maximum fan-out (via a relation r) of a relational structure with domain W is $|W|$ we arrive at the following proposition.

Proposition 4.2 *For any $\mathcal{H}(@,\downarrow)$-formula φ, the following are equivalent:*

(i) *φ is satisfiable.*

(ii) *$FT(\varphi)$ is satisfiable.*

(iii) *$FT(\varphi)^-$ is satisfiable.*

Proof. From the previous discussion, $FT(\varphi)$ is satisfiable iff it is satisfiable by a functional model $\mathcal{I} = \langle W, I, \cdot^{\mathcal{I}}\rangle$ such that $|W| = |I|$. Using any bijection between W and I we define an unsorted model that satisfies $FT(\varphi)^-$. □

Because the number of possible functions of $W \to W$ is $|W|^{|W|}$, this cardinality argument is not compatible with maximal models. However, using classical preservation results we show that one can do with a weaker form of maximality: only the realizations of functions for nodes that are "reachable" from the initial point of evaluation are needed.

Definition 4.3 [Generated submodels] Let $\mathcal{I} = \langle W, \cdot^{\mathcal{I}}\rangle$ and $\mathcal{I}' = \langle W', \cdot^{\mathcal{I}'}\rangle$ be two relational models, and let $g : \mathsf{Nom} \to W$ be a valuation. We say that $\mathcal{I}'$ *is submodel of $\mathcal{I}$ generated by g* whenever:

(i) $\mathit{range}\, g \subseteq W' \subseteq W$,

(ii) if $w \in W'$ then either $w \in \mathit{range}\, g$ or it can be reached from $v \in \mathit{range}\, g$ in a finite number of steps through the relations in the model,

(iii) if $w \in W'$ and $(w, v) \in r^{\mathcal{I}}$ for some $r \in \mathsf{Rel}$, then $v \in W'$,

(iv) $p^{\mathcal{I}'} = p^{\mathcal{I}} \cap W'$ for each $p \in \mathsf{Prop}$,

(v) $r^{\mathcal{I}'} = r^{\mathcal{I}} \cap (W' \times W')$ for each $r \in \mathsf{Rel}$.

Definition 4.4 [g-maximal models] Consider an unsorted functional model $\mathcal{I} = \langle W, \cdot^{\mathcal{I}}\rangle$, and let $r^{\mathcal{I}} \subseteq W \times W$ be the relation induced by $\mathcal{I}$ for each $r \in \mathsf{Rel}$. Let $g : \mathsf{Nom} \to W$ be an assignment and $\mathcal{I}_g = \langle W_g, \cdot^{\mathcal{I}_g}\rangle$ be the generated submodel of $\mathcal{I}$ by g. We say that $\mathcal{I}$ is *g-maximal* if for each function $\alpha : W_g \to W_g$ such that $(v, \alpha(v)) \in r^{\mathcal{I}}$ for all $v \in W_g$, there exists an $i \in W$ for which $f_r^{\mathcal{I}}(i, v) = \alpha(v)$, for all $v \in W_g$.

Theorem 4.5 *Let γ be a prefix-stable formula in the unsorted functional correspondence language with a free variable y that occurs in context $f_r(y, t)$, with* *all the variables in t free in γ. Then, for every g-maximal model $\mathcal{I}$ we have:*

$$\mathcal{I} \models_{\text{FO}} \forall x_1 \dots x_k \exists y.\gamma[g] \textit{ iff } \mathcal{I} \models_{\text{FO}} \exists y \forall x_1 \dots x_k.\gamma[g].$$

The proof is analogous to the one for Theorem 3.5. Proposition 2.5, which states that in the many-sorted case we can assume every satisfiable formula to be satisfied by a maximal model, is a key ingredient in the proof of correctness of the optimized functional translation. For the basic case we have an analogous for g-maximal models:

Proposition 4.6 *A formula φ of $\mathcal{H}(@, \downarrow)$ is satisfiable iff there exists an unsorted functional model $\mathcal{I}$ such that;*

(i) *$\mathcal{I}$ induces a model that satisfies φ at some world w,*

(ii) *$\mathcal{I}$ is g_w-maximal, where g_w is the constant assignment $g_w(x) = w$.*

Proof. We only need to prove the left-to-right direction. Assume, then, that $\mathcal{I}, g, w \models \varphi$ for some $\mathcal{I} = \langle W, \cdot^{\mathcal{I}} \rangle$, $w \in W$ and $g : \mathsf{Nom} \to W$, and pick any unsorted functional model $\mathcal{I}_f = \langle W, \cdot^{\mathcal{I}_f} \rangle$ that induces $\mathcal{I}$. For each $r \in \mathsf{Rel}$ let $\Gamma_r = \{\alpha : W \to W \mid \forall v \,.\, v \not\in de_r^{\mathcal{I}_f} \Rightarrow (v, \alpha(v)) \in r^{\mathcal{I}_f}\}$ and define the set $\Gamma = \bigcup \Gamma_r$. We then construct an unsorted functional model $\mathcal{I}' = \langle W \cup \Gamma, \cdot^{\mathcal{I}'} \rangle$ such that $de_r^{\mathcal{I}'} = de_r^{\mathcal{I}_f} \cup \Gamma$, $p^{\mathcal{I}'} = p^{\mathcal{I}_f}$ for all $p \in \mathsf{Prop}$, and, for every $r \in \mathsf{Rel}$, $f_r^{\mathcal{I}'}$ is an arbitrary function that satisfies $f_r^{\mathcal{I}'}(\alpha, v) = \alpha(v)$ for all $\alpha \in \Gamma_r$ and all $v \in W$. It is straightforward to verify that $\mathcal{I}'$ is g_w-maximal. Moreover, the identity on W is partial isomorphism between $\mathcal{I}$ and the relational model induced by $\mathcal{I}'$, so the latter must also satisfy φ at w. □

Using Proposition 4.6 it is simple to reproduce the proof of Theorem 3.6.

Theorem 4.7 *Let φ be a formula of $\mathcal{H}(@, \downarrow)$. The following are equivalent:*

(i) *φ is satisfiable.*

(ii) *$\exists i{:}\omega.OFT_i(\varphi)$ is satisfiable.*

(iii) *$\exists i.OFT_i(\varphi)^-$ is satisfiable.*

Now, the proof of Proposition 4.6 does not generally work if we are interested in satisfiability for a restricted class of models. For example, it clearly breaks for the class of models defined by the modal axiom $[r]p \to \langle r \rangle p$, since the g_w-maximal model obtained in the proof does not satisfy the seriality condition $\forall x \exists y.r(x, y)$. We will see next that it is possible to obtain a g_w-maximal model that preserves any frame condition that is invariant under disjoint unions. The seriality condition above falls in this category. In fact, by a well-known result due to Goldblatt and Thomason, every class of models that is both first-order and modally definable (that is, definable by a basic modal formula) must be closed under disjoint unions [7].

We begin by defining an operation Ψ_κ on models. Intuitively, $\Psi_\kappa(\mathcal{I})$ is the model obtained by taking κ isomorphic copies of $\mathcal{I}$ (in particular, $\Psi_0(\mathcal{I}) = \mathcal{I}$).

Definition 4.8 Let $\mathcal{I} = \langle W, \cdot^{\mathcal{I}} \rangle$ be a relational model and let κ be an ordinal number; then $\Psi_\kappa(\mathcal{I}) = \langle W^*, \cdot^* \rangle$ is the model such that $W^* = W \cup (\kappa \times W)$, $r^* = r^{\mathcal{I}} \cup \{((a, w), (a, v)) \mid a \in \kappa \text{ and } (w, v) \in r^{\mathcal{I}}\}$ and $p^* = p^{\mathcal{I}} \cup (\kappa \times p^{\mathcal{I}})$.

Clearly, for basic modal formulas, $\mathcal{I}, g, w \models \varphi$ if and only if $\Psi_\kappa(\mathcal{I}), g, w \models \varphi$. Moreover, since $\Psi_\kappa(\mathcal{I})$ is the disjoint union of $\kappa + 1$ copies of $\mathcal{I}$, every class of models $\mathcal{C}$ that is modally definable, is closed by Ψ_κ.

Proposition 4.9 *Let $\mathcal{C}$ be a class of relational models that is closed by Ψ_κ and let φ be a formula of $\mathcal{H}(@, \downarrow)$. Then, φ is $\mathcal{C}$-satisfiable iff there exists an unsorted functional model $\mathcal{I}^*$ such that:*

(i) *$\mathcal{I}^*$ induces a $\mathcal{C}$-model and satisfies φ at some world w,*

(ii) *$\mathcal{I}^*$ is g_w-maximal, where g_w is the constant assignment $g_w(x) = w$.*

Proof. The argument is very similar to that of Proposition 4.6. Given a hybrid model $\mathcal{I} = \langle W, \cdot^{\mathcal{I}} \rangle$ such that $\mathcal{I}, g, w \models \varphi$ for some g and w, we first build the model $\mathcal{I}' = \Psi_\kappa(\mathcal{I})$ with $\kappa = |W|^{|W|}$. By construction, $\mathcal{I}'$ is in $\mathcal{C}$ and as observed above, $\mathcal{I}', g, w \models \varphi$. It is now easy to turn any functional model inducing $\mathcal{I}'$ into a g_w-maximal one. □

Corollary 4.10 *Let $\mathcal{C}$ be a class of modally definable models and let φ be a formula of $\mathcal{H}(@, \downarrow)$. The following are equivalent:*

(i) *φ is $\mathcal{C}$-satisfiable.*

(ii) *$\exists i{:}\omega.OFT_i(\varphi)$ is $\mathcal{C}$-satisfiable.*

(iii) *$\exists i.OFT_i(\varphi)^-$ is $\mathcal{C}$-satisfiable.*

5 General unsoundness of sort erasure for OFT_i

One may wonder if performing sort erasure on the optimized functional translation is sound in the general case, that is, over classes that are not definable in the basic modal language. We give a negative answer to this question by exhibiting a class of models that is definable by a hybrid formula for which sort erasure fails.

Consider the hybrid axiom $\langle s \rangle i$. It is well-known that it defines the class of models that satisfy the first-order formula $\forall xy.s(x, y)$ [5]. Since in this class s behaves like a universal modality, we can use the machinery of hybrid logics to

[5] We can functionally translate the axiom $\langle s \rangle i$ as $\forall x{:}\omega.\neg de_s(x) \wedge \forall xy{:}\omega.\exists j{:}\iota.(x = f_s(j, y))$. This shows that we don't need any special ad-hoc machinery to deal with the universal modality when we use the functional translation.

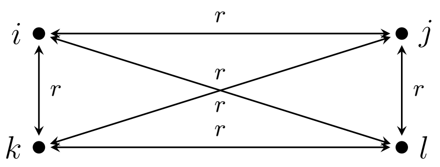


Fig. 3. A $\mathcal{C}$-model for ψ (relation s is omitted).

impose cardinality conditions on models. For example, the following formula is satisfiable only by models with exactly four elements, labelled i, j, k and l:

$$[s](i \vee j \vee k \vee l) \wedge @_i\neg j \wedge @_i\neg k \wedge @_i\neg l \wedge @_j\neg k \wedge @_j\neg l \wedge @_k\neg l \tag{8}$$

Now, let ψ be the conjunction of (8) with the following formulas:

$$\begin{array}{ll} @_i(\langle r\rangle j \wedge \langle r\rangle k \wedge \langle r\rangle l \wedge [r]\neg i) & @_j(\langle r\rangle i \wedge \langle r\rangle k \wedge \langle r\rangle l \wedge [r]\neg j) \\ @_k(\langle r\rangle i \wedge \langle r\rangle j \wedge \langle r\rangle l \wedge [r]\neg k) & @_l(\langle r\rangle i \wedge \langle r\rangle j \wedge \langle r\rangle k \wedge [r]\neg l) \end{array}$$

Clearly, ψ is $\mathcal{C}$-satisfiable. In fact (assuming $\mathsf{Prop} = \emptyset$) any model for ψ is isomorphic to the one in Figure 3. Finally, let φ be the conjunction of:

$$\begin{array}{lll} [s]\langle r\rangle(i \vee j) & [s]\langle r\rangle(i \vee k) & [s]\langle r\rangle(i \vee l) \\ [s]\langle r\rangle(j \vee k) & [s]\langle r\rangle(j \vee l) & [s]\langle r\rangle(k \vee l) \end{array}$$

The model of Figure 3 also satisfies φ and, therefore, satisfies $\psi \wedge \varphi$. Now, consider the formula $OFT_m(\psi \wedge \varphi)$, which has to be satisfiable in $\mathcal{C}$ as well. We will see that the minimum number of elements of sort ι that a $\mathcal{C}$-model for $OFT_m(\psi \wedge \varphi)$ requires is six (in fact, it requires exactly six, but we won't show the upper-bound). Since such a model cannot have more than four elements of sort ω we will conclude that sorts cannot be safely removed in this case.

The first thing to observe is that there is an upper bound for the number of elements of sort ι required, given by the number of existentially quantified variables in $OFT_m(\psi \wedge \varphi)$. The latter is the number of diamonds in $\psi \wedge \varphi$, namely, eighteen. We need a function to witness each of these existentially quantified variables, and the minimum number of elements of sort ι is simply the minimum number of distinct functions required.

We restrict our attention to diamonds in φ. Exactly one diamond occurs in each conjunct; informally, we say that a function is required to account for each of them. In the end, we conclude that six distinct functions are needed.

Let $\mathcal{I} = \langle W, \cdot^{\mathcal{I}}\rangle$ be the model of Figure 3 and assume $W = \{i, j, k, l\}$. Moreover, assume $g(w) = w$ for $w \in W$. Consider the conjunct $[s]\langle r\rangle(i \vee j)$; since $[s]$ is a universal modality, we have have $\mathcal{I}, g, w \models \langle r\rangle(i \vee j)$ for every $w \in W$. Let α_1 be the function witnessing the diamond. Then $\mathcal{I}, g, \alpha_1(w) \models (i \vee j)$. Since $r^{\mathcal{I}}$ is irreflexive, α_1 satisfies $\alpha_1(i) = j$ and $\alpha_1(j) = i$. We can do this analysis for all six conjuncts of φ; the constraints are shown in Table 1.

$\alpha_1(i) = j$	$\alpha_1(j) = i$	$\alpha_1(k) \in \{i, j\}$	$\alpha_1(l) \in \{i, j\}$
$\alpha_2(i) = k$	$\alpha_2(j) \in \{i, k\}$	$\alpha_2(k) = i$	$\alpha_2(l) \in \{i, k\}$
$\alpha_3(i) = l$	$\alpha_3(j) \in \{i, l\}$	$\alpha_3(k) \in \{i, l\}$	$\alpha_3(l) = i$
$\alpha_4(i) \in \{j, k\}$	$\alpha_4(j) = k$	$\alpha_4(k) = j$	$\alpha_4(l) \in \{j, k\}$
$\alpha_5(i) \in \{j, l\}$	$\alpha_5(j) = l$	$\alpha_5(k) \in \{j, l\}$	$\alpha_5(l) = j$
$\alpha_6(i) \in \{k, l\}$	$\alpha_6(j) \in \{k, l\}$	$\alpha_6(k) = l$	$\alpha_6(l) = k$

Table 1
Requirements for the functions that satisfy $OFT_m(\varphi)$.

Because α_1, α_2 and α_3 differ in the value for i, they must be all distinct functions. Similarly, α_4 differs from α_1 in the value for j, from α_2 in the value for k and from α_3 everywhere. From a similar analysis for α_5 and α_6 we conclude that six distinct functions are needed to satisfy these six formulas.

6 Conclusions and future work

When dealing with functional translations, many-sorted first-order logic is undeniably useful for presentation reasons. In this article we discussed in which cases many-sorted logic is needed also for technical reasons. We proved that as long as reasoning is confined to classes of models closed by disjoint unions (e.g., modally definable classes) sorts can be eliminated.

It was shown in [14] that the optimized functional translation can be used also to reason over some modally defined classes that are not first-order, like the class defined by the McKinsey axiom $\Box\Diamond p \rightarrow \Diamond\Box p$. It is easily seen that sorts are not required in that case either.

Of course, the empirical advantages of eliminating sorts need to be assessed. One could in principle pick an off-the-shelf automated prover and benchmark their performance on a number of functionally translated formulas (generated at random or from a given domain), both with and without sort annotations. However, it is not at all clear if one is warranted to extract meaningful conclusions from this sort of black-box experiments. E.g., no noticeable differences may be due to a bottleneck in the clausification process, or even a prover implementing an heuristic that amounts to erasing sorts; better execution times for the unsorted case may be due to a deficient handling of larger formulas, etc.

This article shows that the functional translation adapts surprisingly well to the hybrid case (see, for comparison, the case of the layered translation in [8]). Schmidt established in [16] that, when restricted to the basic modal case, any refinement of resolution plus the (eagerly applied) condensing rule [11] is terminating for the output of the optimized functional translation. Most first-order theorem provers have factoring and subsumption deletion rules, and hence condensing is in fact implicit when the implementation is fair. This means that any standard (complete and fair) resolution theorem prover used along with the optimized translation constitutes a decision method for the basic modal language over the class of all models. Termination conditions for some frame classes were also investigated. As future work, it would be interesting to see if termination can be achieved in the case of $\mathcal{H}(@)$.

References

[1] C. Areces, R. Gennari, J. Heguiabere, and M. de Rijke. Tree-based heuristics in modal theorem proving. In *Proc. of ECAI'2000*, pages 199–203, 2000.

[2] C. Areces and B. ten Cate. Hybrid logics. In *Handbook of Modal Logics*, pages 821–868. Elsevier, 2006.

[3] Y. Auffray and P. Enjalbert. Modal theorem proving: An equational viewpoint. In *Proc. of the 11th IJCAI*, pages 441–445. Morgan Kaufmann Pub., 1989.

[4] Y. Auffray and P. Enjalbert. Modal theorem proving: An equational viewpoint. *J. of Logic and Computation*, 2(3):247–295, 1992.

[5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2002.

[6] L. Fariñas del Cerro and A. Herzig. Linear modal deductions. In *Proc. of CADE-9*, volume 310 of *LNCS*, pages 487–499. Springer, 1988.

[7] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 163–173. Springer, 1975.

[8] D. Gorín. *Automated reasoning techniques for hybrid logics.* PhD thesis, Universidad de Buenos and Université Henri Poincaré, 2009.

[9] I. Horrocks, U. Hustadt, U. Sattler, and R. Schmidt. Computational modal logic. In *Handbook of Modal Logics*, pages 181–245. Elsevier, 2006.

[10] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *J. of Applied Non-Classical Logics*, 9(4):479–522, 1999.

[11] W. Joyner, Jr. Resolution strategies as decision procedures. *J. of the ACM*, 23(3):398–417, 1976.

[12] H. Ohlbach. *A Resolution Calculus for Modal Logics.* PhD thesis, Universität Kaiserslautern, 1988.

[13] H. Ohlbach. A resolution calculus for modal logics. In *Proc. of CADE-9*, volume 310 of *LNCS*, pages 500–516. Springer, 1988.

[14] H. Ohlbach and R. Schmidt. Functional translation and second-order frame properties of modal logics. *J. of Logic and Computation*, 7(5):581–603, 1997.

[15] R. Schmidt. *Optimised Modal Translation and Resolution.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.

[16] R. Schmidt. Decidability by resolution for propositional modal logics. *J. of Automated Reasoning*, 22(4):379–396, 1999.

[17] C. Walther. Many-sorted inferences in automated theorem proving. In *Sorts and Types in Artificial Intelligence*, volume 418 of *LNCS*, pages 18–48. Springer, 1989.

[18] C. Weidenbach. Combining superposition, sorts and splitting. In *Handbook of Automated Reasoning*, volume 2, chapter 27, pages 1965–2014. Elsevier and MIT Press, 2001.

[19] C. Weidenbach, R. Schmidt, T. Hillenbrand, R. Rusev, and D. Topic. System description: Spass version 3.0. In *Proc. of CADE-21*, number 4603 in LNAI, pages 514–520. Springer-Verlag, 2007.

[20] N. Zamov. Modal resolutions. *Soviet Math*, 33(9):23–29, 1989.