**FMLtoHOL (version 1.0): Automating First-order Modal Logics with LEO-II and Friends**

Christoph Benzmüller¹ and Thomas Raths²

¹ Freie Universität Berlin, Germany
² University of Potsdam, Germany

**Abstract.** A converter from first-order modal logics to classical higher-order logic is presented. This tool enables the application of off-the-shelf higher-order theorem provers and model finders for reasoning within first-order modal logics. The tool supports logics K, K4, D, D4, T, S₄, and S₅ with respect to constant, varying and cumulative domain semantics.

### 1 Introduction

First-order modal logics (FMLs) [4] have many applications, e.g., in planning, natural language processing, program verification, querying knowledge bases, and modeling communication. These applications motivate the use of automated theorem proving (ATP) systems for FMLs. Several new FML ATP systems, including two FMLtoHOL-based solutions, have recently been provided [1].

This paper describes the FMLtoHOL tool, which automatically converts problems in FML, formulated in the new qmf-syntax [5] (which extends the TPTP fol-syntax [7] with operators #box and #dia), into problems in classical higher-order logic (HOL), formulated in thf₀-syntax [6]. FMLtoHOL exploits and implements a semantic embedding of constant domain FML in HOL [2]. Moreover, the tool extends this embedding to varying and cumulative domain semantics.

FMLtoHOL thus turns any thf₀-compliant HOL ATP system — such as LEO-II³ and Satallax³ — into a flexible ATP system for FML. At present FMLtoHOL supports modal logics $L := \{\text{K, K4, D, D4, T, S₄, S₅}\}$, all with respect to constant, varying and cumulative domain semantics. Extending the tool to other normal FMLs and their combinations is straightforward.

In the remainder the language of FML is fixed as: $F, G := P(t₁, \ldots, tₙ) \mid \neg F \mid F \land G \mid F \lor G \mid F \Rightarrow G \mid \Box F \mid \Diamond F \mid \forall x F \mid \exists x F$. The symbols $P$ are $n$-ary ($n \geq 0$) relation constants which are applied to terms $t₁, \ldots, tₙ$. The $tᵢ$ ($0 \leq i \leq n$) are ordinary first-order terms and they may contain function symbols. The formula $(\forall x \Box fx) \Rightarrow \Box \forall x fx$ is used as an example, it is referred to as E1. In constant domain (resp. varying domain) semantics E1 is a theorem (resp. countersatisfiable) for logics $L$. In cumulative domain semantics E1 is a theorem for S₅ and countersatisfiable for the other logics in $L$.

---

* Supported by the German Research Foundation (BE2501/9-1 and KR858/9-1).
³ Cf. [www.leoprover.org](http://www.leoprover.org) and [www.ps.uni-saarland.de/~cebrown/satallax/](http://www.ps.uni-saarland.de/~cebrown/satallax/)
2 Theory of FMLtoHOL

FMLtoHOL exploits the fact that Kripke structures can be elegantly embed-
ded in HOL [23]: FML propositions $F$ are associated with HOL terms $F_\rho$ of
predicate type $\rho := t \mapsto o$. Type $o$ denotes the set of truth values and type $t$
is associated with the domain of possible worlds. Thus, the application $(F_\rho w_i)$
corresponds to the evaluation of FML proposition $F$ in world $w_i$. Consequently,
validity is formalized as $\text{vld}_{\rho \rightarrow o} = \lambda F_\rho \forall w F w$. Classical connectives like $\neg$ and
$\vee$ are simply lifted to type $\rho$ as follows: $\neg F_\rho = \lambda F_\rho \lambda w_1 \neg F w_1$ and $\vee F_\rho = \lambda F_\rho \lambda G_\rho \lambda w_1 (F w_1 \vee G w_1)$. $\Box$ is modeled as $\Box F_\rho = \lambda F_\rho \lambda w_1 (\forall v \Box F v)$, where
constant symbol $R_{\rho \rightarrow o}$ denotes the accessibility relation of the $\Box$ operator, which
remains unconstrained in logic $K$. Further logical connectives are defined as
usual: $\wedge = \lambda F_\rho \lambda G_\rho \neg(\neg F \vee \neg G)$, $\Rightarrow = \lambda F_\rho \lambda G_\rho (\neg F \vee G)$, $\Diamond = \lambda F_\rho \neg \Box \neg F$. To obtain e.g. modal logics $D$, $T$, $S4$, and $S5$, $R$ is axiomatized as serial, reflexive,
reflexive and transitive, and an equivalence relation, respectively. Arbitrary
normal modal logics extending $K$ can be axiomatized this way.

For individuals a further base type $\mu$ is reserved in HOL. Universal quantifica-
tion $\forall x F$ is introduced as syntactic sugar for $\Pi \lambda x F$, where $\Pi$ is defined
as follows: $\Pi_{(\mu \rightarrow o)} = \lambda H_{\mu \rightarrow o} \lambda w_1 \forall x_\mu H x w_1$. For existential quantification,
$\exists x F$ is then syntactic sugar for $\Sigma \lambda x F$. $n$-ary relation symbols $P$, $n$-ary function symbols $f$ and individual constants $c$ in FML
obtain types $\mu_1 \rightarrow \ldots \rightarrow \mu_n \rightarrow \rho$, $\mu_1 \rightarrow \ldots \rightarrow \mu_n \rightarrow \mu_{n+1}$ (both with $\mu_i = \mu$ for
$0 \leq i \leq n + 1$) and $\mu$, respectively.

For any FML formula $F$ holds: $F$ is a valid in modal logic $K$ for constant
domain semantics if and only if $\text{vld}_{\rho \rightarrow o} F_\rho$ is valid in HOL for Henkin semantics.
This correspondence provides the foundation for proof automation of FMLs with
HOL-ATP systems. The correspondence follows from [2], where a more general
result is shown for FMLs with additional quantification over Boolean variables.

The above approach is adopted for varying domain semantics as follows: 1.
$\Pi$ is now defined as $\Pi = \lambda H_{\mu \rightarrow o} \lambda w_1 \forall x_\mu \text{exInW} x w \Rightarrow H x w$, where relation
$\text{exInW}_{\mu \rightarrow o}$ (for ‘exists in world’) relates individuals with worlds. 2. The non-
emptiness axiom $\forall w_1 \exists x_\mu \text{exInW} x w$ for these individual domains is added. 3. For
each individual constant symbol $c$ an axiom $\forall w_1, \text{exInW} x w$ is postulated; these
axioms enforce the designation of $c$ in the individual domain of each world $w$.
Analogous designation axioms are required for function symbols.

For cumulative domain semantics the axiom $\forall x_\mu \forall v_1, \forall v_2, \text{exInW} x w \wedge R x w \Rightarrow \text{exInW} x w$ is additionally postulated. It states that the individual domains are
increasing along relation $R$.

3 Implementation and Functionality of FMLtoHOL

FMLtoHOL is implemented as part of the TPTP2X tool [7]. TPTP2X is a multi-
functional utility for generating, transforming, and reformattting TPTP problem
files. It is written in Prolog and it can be easily modified and extended.

The tool is invoked as “tptp2X -f thf:<logic>::<domain> <qmf-file>”
where $<logic> \in \{K,K4,D,D4,T,S4,S5\}$ and $<domain> \in \{\text{const}, \text{vary}, \text{cumul}\}$.

To illustrate its use it is assumed that file $E1.qmf$ contains $E1$ in $\text{qmf}$-syntax:
qmf(con,conjecture,(  
  ( ! [X] : ( #box : ( f(X) ) ) ) => ( #box : ( ! [X] : ( f(X) ) ) ) ).

"tptp2X -f thf:d:const E1.qmf" generates the corresponding HOL problem file E1.thf in thf-syntax for constant domain logic D:

%----Include axioms for modal logic D under constant domains
include('Axioms/LCL013^0.ax.const').

%------------------------------------------------------------------------

thf(f_type,type,( f: mu > $i > $o )). % type declaration for constant f

thf(prove,conjecture,( mvalid @  
  ( mimplies @ ( mforall_ind @ ^ 
    [X: mu] : ( mbox_d @ ( f @ X ) ) )  
  @ ( mbox_d @ ( mforall_ind @ ^ 
      [X: mu] : ( f @ X ) ) ) ) )).

The included axiom files contain the definitions of the logical connectives as outlined in Sect. 2. For example, the definition for mforall_ind (which realizes \( \Pi \) for constant domain semantics) is given in LCL013^0.ax.const:

thf(mforall_ind,definition,( mforall_ind =  
  ( ^ 
    [Phi: mu > $i > $o,W: $i] : ! 
      [X: mu] : ( Phi @ X @ W ) ) )).

LCL013^2.ax contains the definition of the serial \( \Box \) operator in logic D:

thf(mbox_d,definition,( mbox_d =  
  ( ^ 
    [Phi: $i > $o,W: $i] : ! 
      [V: $i] : ( ~ ( rel_d @ W @ V ) | ( Phi @ V ) ) ) )).

thf(a1,axiom,( mserial @ rel_d )).

Similar definitions are provided in the included axiom files for the other logical connectives and for auxiliary terms like mserial. The HOL ATP systems LEO-II and Satallax when applied to E1.thf find a proof within a few milliseconds.

When FMLtoHOL is called with option "-f thf:s5:vary" a modified file E1.thf is created containing a conjecture identical to above except that mbox_d is replaced by mbox_s5. Moreover, E1.thf now includes different axiom files LCL013^0.ax.vary and LCL013^6.ax. The former contains a modified definition of mforall_ind, which incorporates an explicit 'exists in world' condition:

thf(mforall_ind,definition,( mforall_ind =  
  ( ^ 
    [Phi: mu > $i > $o,W: $i] : ! 
      [X: mu] : ( ( exists_in_world @ X @ W ) => ( Phi @ X @ W ) ) ) )).

thf(nonempty_ax,axiom,(  
  ! [V : $i] : ? 
    [X : mu] : (exists_in_world @ X @ V) ).

The latter axiom specifies the domains of existing objects as non-empty for all worlds worlds V. Axiom file LCL013^6.ax specifies mbox_s5 as follows:

4 Some explanations: " is \( \lambda \)-abstraction and \( \circ \) an (explicit) application operator. !, ?, ~, |, and => encode universal and existential quantification, negation, disjunction and implication in HOL. mu > $i > $o encodes the HOL type \( \mu \rightarrow o \). mimplies, mforall_ind, and mbox_d are embedded logical connectives as described in Sect. 2. Their denotation is fixed by adding definition axioms; see e.g. mforall_ind below.
thf(mbox_s5,definition,( mbox_s5 =
  ( ~ [Phi: $i > $o,W: $i] :
  ! [V: $i] : ( ~ ( rel_s5 @ W @ V ) | ( Phi @ V ) ) ) )).

thf(a1,axiom,( mreflexive @ rel_s5 )).

thf(a2,axiom,( mtransitive @ rel_s5 )).

thf(a3,axiom,( msymmetric @ rel_s5 )).

For the modified problem Satallax finds a counter model within milliseconds.

4 Discussion and Outlook

The FMLtoHOL has been applied and evaluated in combination with the HOL ATP systems Satallax and LEO-II; cf. [1] for details. In this case study the approach has also been compared with other, heterogeneous FML ATP systems. The FMLtoHOL based solution has the best coverage (and it can easily be extended to other modal logics and their combinations) and it is second best in overall performance behind the clausal connection prover MeanCoP.

Future work includes several optimizations of the tool, extensions for multimodal logics (which it already partly supports), and further case studies. These case studies should evaluate the tool also in combination with other thf0-compliant HOL provers and model finders as outlined in [6]: TPS, Isabelle, Refute and Nitpick.

A recent observation is that the HOL model finders Satallax, Refute and Nitpick apparently work well for constant and varying domain semantics but have problems to find counter models for cumulative domain semantics.

Acknowledgements: We thank Jens Otten, Geoff Sutcliffe and Chad Brown for their support and their valuable input to this work.

References

1. C. Benzmüller, T. Raths and J. Otten. Implementing and Evaluating Provers for First-order Modal Logics, Proceedings of ECAI’2012, to appear.
2. C. Benzmüller and L. C. Paulson. Quantified Multimodal Logics in Simple Type Theory. Logica Universalis, 2012. DOI:10.1007/s11787-012-0052-y
3. C. Benzmüller and L. C. Paulson, Exploring Properties of Normal Multimodal Logics in Simple Type Theory with LEO-II. In Festschrift in Honor of Peter B. Andrews on His 70th Birthday, 2008. College Publications.
4. M. Fitting and R. L. Mendelsohn. First-Order Modal Logic. Kluwer, 1998.
5. T. Raths and J. Otten. The QMLTP Problem Library for First-order Modal Logics. The 6th International Joint Conference on Automated Reasoning (IJCAR), Manchester, UK, 2012.
6. G. Sutcliffe and C. Benzmüller. Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. Journal of Formalized Reasoning, 3(1):1-27, 2010.
7. G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. Journal of Automated Reasoning, 43(4):337–362, 2009.

---

[5] Cf. the information at [http://www.iltp.de/qmltp/systems.html](http://www.iltp.de/qmltp/systems.html)