HOL based First-order Modal Logic Provers*

Christoph Benzmüller[1] and Thomas Raths[2]

[1] Dep. of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Institute for Computer Science, University of Potsdam, Germany

Abstract. First-order modal logics (FMLs) can be modeled as natural fragments of classical higher-order logic (HOL). The FMLtoHOL tool exploits this fact and it enables the application of off-the-shelf HOL provers and model finders for reasoning within FMLs. The tool bridges between the `qmf`-syntax for FML and the TPTP `thf0`-syntax for HOL. It currently supports logics K, K4, D, D4, T, S4, and S5 with respect to constant, varying and cumulative domain semantics. The approach is evaluated in combination with a meta-prover for HOL, which sequentially schedules various HOL reasoners. The resulting system is very competitive.

1 Introduction

First-order modal logics (FMLs) [7] have many applications and these applications motivate the use of automated theorem proving systems for FMLs. Until recently no (correct) ATP systems for FMLs were available.[3] However, good progress has been made in the last two years, and novel provers have recently been implemented and compared [1]. Among these systems is also an approach based on classical higher-order logic (HOL) [3, 2]. This HOL approach, which is further improved and evaluated here, is the focus of this paper. The particular contributions include:

(A) The FMLtoHOL tool is presented, which converts problems in FML, formulated in `qmf`-syntax [13] (which extends the TPTP `fol`-syntax [15] with operators `#box` and `#dia`), into HOL problems in `thf0`-syntax [16].[4] FMLtoHOL implements a semantic embedding of constant domain FMLs in HOL [3]. The tool has been extended to also support varying and cumulative domains. FMLtoHOL turns any `thf0`-compliant HOL ATP system into a flexible ATP system for FMLs. At present FMLtoHOL supports modal logics from $L := \{K, K4, D, D4, T, S4, S5\}$. However, its extension to further normal FMLs is straightforward.

(B) The FMLtoHOL tool is exemplarily applied in combination with a meta-prover for HOL, called HOL-P in the remainder. This meta-prover exploits the

* Supported by the German Research Foundation (grants BE2501/9-1 & KR858/9-1).

[3] A pioneering prover is GQML [17]. However, GQML has been excluded in recent experiments (in [1] or here) since it returned incorrect results for several formulae.

[4] `thf` stands for *typed higher-order form* and it refers to family of syntax formats for higher-order logic. So far only the fully developed `thf0` format, for simply typed lambda calculus, is in practical use.

SystemOnTPTP infrastructure [15] and sequentially schedules the HOL reasoners LEO-II [4], Satallax [6], Isabelle [10], agsyHOL [9] and Nitpick [5]. HOL-P is evaluated with respect to 580 benchmark problems in the QMLTP library [13]. As a side contribution a complete translation of the QMLTP library (for all logics in L, all different domain conditions, and both options as explained in (C)) into HOL (resp. `thf0`) is achieved, resulting in $7 \times 3 \times 2 \times 580 = 24360$ problems. The 3480 problems for logic S4 can be download from http://christoph-benzmueller.de/papers/THF-S4-ALL.zip; others can be requested by EMail.

(C) There are different options in the HOL approach for the modeling of logics in L. One is to state the conditions on the accessibility relation R associated with $\Box$ 'semantically', e.g, $\forall x \exists y Rxy$ expresses that R is serial. Exploiting quantification over booleans ($\forall^p$) (cf. [3]) the corresponding 'syntactical' axiom $\forall^p p(\Box p \Rightarrow \Diamond p)$ may instead be postulated. FMLtoHOL so far only supports the 'semantical' approach. A first evaluation of both options is provided in this paper. To enable this the semantical example problems have been converted into their syntactical counterparts by hand.

The structure of the paper is as follows: §2 outlines FML. §3 and §4 describe the theory and implementation of FMLtoHOL. §5 introduces prover HOL-P. Experiments are presented in §6, and §7 concludes the paper.

2 First-Order Modal Logic

The syntax of FML adopted in this paper is: $F, G ::= P(t_1, \ldots, t_n) \mid \neg F \mid F \wedge G \mid F \vee G \mid F \Rightarrow G \mid \Box F \mid \Diamond F \mid \forall x F \mid \exists x F$. The symbols P are n-ary ($n \geq 0$) relation constants which are applied to terms $t_1, \ldots, t_n$. The t_i ($0 \leq i \leq n$) are ordinary first-order terms and they may contain function and constant symbols. The usual precedence rules for logical constants are assumed. The formula E1 $:= (\Diamond \exists x Pfx \wedge \Box \forall y (\Diamond Py \Rightarrow Qy)) \Rightarrow \Diamond \exists z Qz$ is used as a running example.

Regarding semantics, a Kripke style semantics for FML is adopted [7]. In particular, it is assumed that constants and terms are denoting and rigid, i.e. they always pick an object and this pick is the same object in all worlds. Regarding the universe of discourse constant domain, varying domain and cumulative domain semantics are considered. With respect to these base choices the normal modal logics K, K4, K5, B, D, D4, T, S4, and S5 are studied.

3 Theory of FMLtoHOL

FMLtoHOL exploits the fact that Kripke structures can be elegantly embedded in HOL [3]: FML propositions F are associated with HOL terms F_ρ of predicate type $\rho := \iota \to o$. Type o denotes the set of truth values and type ι is associated with the domain of possible worlds. Thus, the application $(F_\rho w_\iota)$ corresponds to the evaluation of FML proposition F in world w. Consequently, validity is formalized as $vld_{\rho \to o} = \lambda F_\rho \forall w_\iota F w$. Classical connectives like $\neg$ and $\vee$ are simply lifted to type ρ as follows: $\neg_{\rho \to \rho} = \lambda F_\rho \lambda w_\iota \neg F w$ and $\vee_{\rho \to \rho \to \rho} = \lambda F_\rho \lambda G_\rho \lambda w_\iota (Fw \vee Gw)$. $\Box$ is modeled as $\Box_{\rho \to \rho} = \lambda F_\rho \lambda w_\iota \forall v_\iota (\neg Rwv \vee Fv)$, where constant symbol $R_{\iota\to\rho}$ denotes the accessibility relation of the $\Box$-operator, which remains unconstrained in logic K. Further logical connectives are defined as usual: $\wedge = \lambda F_\rho \lambda G_\rho \neg(\neg F \vee \neg G)$, $\Rightarrow = \lambda F_\rho \lambda G_\rho (\neg F \vee G)$, $\Diamond = \lambda F_\rho \neg\Box\neg F$.

For individuals a further base type μ is reserved in HOL. Universal quantification $\forall x F$ is introduced as syntactic sugar for $\Pi \lambda x F$, where constant Π is defined as follows: $\Pi_{(\mu\to\rho)\to\rho} = \lambda H_{\mu\to\rho} \lambda w_\iota \forall x_\mu H x w$. For existential quantification, $\Sigma = \lambda H_{\mu\to\rho} \neg \Pi \lambda x_\iota \neg H x$ is introduced. $\exists x F$ 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 $\mu_1 \to \ldots \to \mu_n \to \rho$, $\mu_1 \to \ldots \to \mu_n \to \mu_{n+1}$ (both with $\mu_i = \mu$ for $0 \leq i \leq n+1$) and μ, respectively.

Moreover, universal quantification over propositional variables is added. Similar to above this can be done by introducing a constant Π^p. Π^p and Π are similar and only differ wrt the argument type: $\Pi^p_{(\rho\to\rho)\to\rho} = \lambda H_{\rho\to\rho} \lambda w_\iota \forall p_\rho H p w$. Again, $\forall^p p F$ is introduced as syntactic sugar for $\Pi^p \lambda p F$, etc.

For any FML formula F holds: F is a valid in modal logic K for constant domain semantics if and only if $vld\, F_\rho$ is valid in HOL for Henkin semantics. This correspondence provides the foundation for proof automation of FMLs with HOL-ATP systems. The correspondence is shown in [3].

To extend the above result for logic K to modal logics K4, K5, B, D, D4, T, S4, S5 etc., one may choose between a ‘syntactical’ and a ‘semantical’ approach: (**Semantical**) Axioms such as $\forall x R x x$ or $\forall x \forall y \forall z (Rxy \wedge Ryz \Rightarrow Rxz)$ are postulated to ensure that accessibility relation R obeys certain restrictions, here reflexivity and transitivity. (**Syntactical**) Propositional quantification is exploited to postulate corresponding axioms such as $\forall^p p(\Box p \Rightarrow p)$ or $\forall^p p(\Box p \Rightarrow \Box\Box p)$. These axioms characterize R as reflexive and transitive. Similar axioms exist for other FMLs. Respective correspondences between semantical properties of R and respective syntactical axioms are well known.

Arbitrary normal modal logics extending K can be axiomatized this way. There are cases where only the semantical approach is applicable. For example, irreflexivity of accessibility relation R cannot be axiomatized in the syntactic approach. However, it can trivially be modeled in the semantic approach. In other cases the syntactical approach appears more suitable. Examples are non-Stahlquist formulas like the Löb axiom or the McKinsey formula, for which there are no corresponding first-order semantical conditions on R. Note, however, that the HOL approach is not restricted to first-order conditions on R.

The above approach realizes constant domain semantics. For varying domain semantics it is modified: (1) Π is defined as $\Pi = \lambda H_{\mu\to\rho} \lambda w_\iota \forall x_\mu (\mathtt{exInW} x w \Rightarrow H x w)$, where relation $\mathtt{exInW}_{\mu\to\iota\to o}$ (for ‘exists in world’) relates individuals with worlds. (2) The non-emptiness axiom $\forall w_\iota \exists x_\mu \mathtt{exInW} x w$ for these individual domains is added. (3) For each individual constant symbol c an axiom $\forall w_\iota \mathtt{exInW} c w$ is postulated; these axioms enforce the designation of c in the individual domain of each world w. Analogous designation axioms are added for function symbols. For cumulative domains the axiom $\forall x_\mu \forall v_\iota \forall w_\iota (\mathtt{exInW} x v \wedge R v w \Rightarrow \mathtt{exInW} x w)$ is additionally postulated. It states that the individual domains are increasing along accessibility relation R.

4 Implementation and Functionality of FMLtoHOL

FMLtoHOL is implemented as part of the TPTP2X tool [15], and it is included in the QMLTP—v1.1 package.[5] 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>` $\in \{const, vary, cumul\}$.

Assume that file `E1.qmf` contains example problem E1 in `qmf`-syntax:

```
qmf(con,conjecture,
    ( ((#dia:  ? [X] : p(f(X))) & (#box: ! [Y]: ((#dia: p(Y)) => q(Y))))
      => #dia: ? [Z] : q(Z) )).
```

The command '`./tptp2X -f thf:d:const E1.qmf`' generates a corresponding HOL problem file `E1.thf` in `thf0`-syntax[6] [16] for constant domain logic D:

```
%----Include axioms for modal logic D under constant domains
include('Axioms/LCL013^0.ax.const').
include('Axioms/LCL013^2.ax').

%------------------------------------------------------------------------------
thf(q_type,type,( q: mu > $i > $o )).
thf(p_type,type,( p: mu > $i > $o )).
thf(f_type,type,(f: mu > mu )).

thf(con,conjecture, ( mvalid @
    ( mimplies @
      ( mand @
        ( mdia_d @ ( mexists_ind @ ^ [X: mu] : ( p @ ( f @ X ) ) ) ) @
        ( mbox_d @ ( mforall_ind @ ^ [Y: mu] :
            ( mimplies @ ( mdia_d @ ( p @ Y ) ) @ ( q @ Y ) ) ) ) ) ) @
      ( mdia_d @ ( mexists_ind @ ^ [Z: mu] : ( q @ Z ) ) ) ) )).
```

`mimplies`, `mand`, `mbox_d`, etc. should be read as 'modal-implies', 'modal-and', 'modal-box-d', respectively. The included axiom files contain the definitions of these connectives as outlined in §2. E.g., the definition for `mforall_ind` (which realizes Π 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 ) ) )).
```

File `LCL013^2.ax` contains the definition of the serial □-operator in logic D:

[5] The QMLTP library is available online at http://www.iltp.de/qmltp/problems.html.

[6] Some explanations: `^` is λ-abstraction and `@` an (explicit) application operator. `!`, `?`, `~`, `|`, and `=>` encode universal and existential quantification, negation, disjunction and implication in HOL. `mu > $i > $o` encodes the HOL type $\mu \to \iota \to o$. `mimplies`, `mforall_ind`, and `mbox_d` are embedded logical connectives as described in §2. Their denotation is fixed by adding definition axioms; see e.g. `mforall_ind` below.

```
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`. For problem `E1.thf` Nitpick finds a countermodel in 8 seconds (when run with a 20s time limit).

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` and `rel_d` by `rel_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`, adds a non-emptiness axiom, and adds further axioms as required (cf. conditions (1)-(3) in §3). Axiom file `LCL013^6.ax` specifies `mbox_s5` as follows:

```
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 )).
```

The modified problem in file `E1.thf` is proved by Satallax and LEO-II within milliseconds.

The above explanations are all with respect to the adapted `tptp2X` command that comes with the QMLTP package. The included axiom files, like LCL013ˆ6.ax etc., are also provided by this package, so that only the QMLTP package is required for installing the FMLtoHOL tool.

5 The Prover **HOL-P**

In the experiments the following HOL provers were applied: Satallax (2.6) [6], Isabelle (2012) [10], LEO-II [4] (1.5.0), Nitpick (2012) [5] and agsyHOL (1.0) [9]. Isabelle, Satallax, LEO-II and agsyHOL are theorem provers. Nitpick is a (counter-) model finder. Satallax, and to a lesser extend LEO-II, are also capable of finding countermodels. These systems work for Henkin semantics and they support the `thf0`-syntax as a common input language. Moreover, the SystemOnTPTP infrastructure [15] enables remote calls to instances of these provers at the University of Miami (running on 2.80GHz computers with 1GB memory). Exploiting these features, a simple shell script has been written that bundles these systems into a HOL meta-prover, called HOL-P in the remainder. HOL-P has been employed in the experiments. Using the SystemOnTPTP infrastructure the experiments below can be easily replicated.

6 Evaluation

The QMLTP library [13] is a benchmark library for testing and evaluating ATP systems for FML. It is similar to the TPTP library for classical logic [15] and the ILTP library for intuitionistic logic [14]. Version 1.1 of the QMLTP library includes 600 FML problems divided into 11 problem domains. The problems were taken from different applications, various textbooks, and Gödel's embedding of intuitionistic logic. It also includes 20 problems in multimodal logic. Only the HOL approach is applicable to them do date. Therefore these multimodal logic problems have not been included in our experiments.

HOL-P has been applied in several experiment runs to all 580 monomodal problems in the QMLTP library. The overall time limit of 600s for each problem was equally distributed over the five subprovers of HOL-P. Thus, each subprover was given a 120s time limit per problem. In each experiment run, a different setting with respect to the selected logic (here D and S4) and the domain condition (constant, cumulative, varying) was chosen. The results for HOL-P are presented in Table 1. Moreover, in Table 1 the performance of HOL-P is compared to corresponding results as reported on the QMLTP-website[7] for the provers f2p-MSPASS—3.0 (an instance-based prover which employs MSPASS [8] to prove or refute the propositional formulas it generates), MleanSeP—1.2 (a sequent prover; its calculus extends the classical sequent calculus with specific rules for □ and ◇), MleanTAP—1.3 (a tableaux prover; a classical tableaux calculus is extended by adding and employing prefixes to each formula), and MleanCoP—1.2 (a connection prover based on leanCoP [12, 11]; again formula prefixes are employed). Previous results on the HOL provers LEO-II and Satallax (cf. [1]) have not been included in Table 1; they are now subsumed by HOL-P which is significantly stronger than both of them.

The HOL approach has the broadest coverage of logics and domain conditions (and, as mentioned before, it can easily be adapted to support further logics):

ATP system	supported modal logics	supported domain cond.
MleanSeP 1.2	K,K4,D,D4,T,S4	constant,cumulative
MleanTAP 1.3	D,T,S4,S5	constant,cumulative,varying
MleanCoP 1.2	D,T,S4,S5	constant,cumulative,varying
f2p-MSPASS 3.0	K,K4,K5,B,D,T,S4,S5	constant,cumulative
HOL-P	K,K4,K5,B,D,D4,T,S4,S5	constant,cumulative,varying

The experiments show that the HOL approach is very competitive. In particular, with respect to the accumulated numbers of solved problems in each category HOL-P has a slight lead (HOL-P solved 2225 problems, MleanCoP 2129). This is due to the excellent performance of the (counter-)model finder Nitpick (which fully subsumes Satallax in the Non-Theorem-category of the experiments and beats MleanCoP by quite a margin). In both categories, Theorems and Non-Theorems, HOL-P solved many problems whose QMLTP status was 'Unsolved'.

[7] Cf. http://www.iltp.de/qmltp/download/QMLTP-v1.1-comparison.txt

Table 1. No. of proved monomodal problems (for constant/cumulative/varying domain semantics, in this order) of the QMLTP library. All provers were run with a 600s time limit. In HOL-P a timeout of 120s was given to each subprover.

	MleanSeP	MleanTAP	f2p-MSPASS	MleanCoP	HOL-P
Logic D: constant/cumulative/varying domains					
Theorem	135/130/–	134/120/100	076/079/–	**217/200/170**	208/184/163
Non-Thm	001/004/–	004/004/004	107/108/–	209/204/243	**250/269/295**
Solved	136/134/–	138/124/104	183/187/–	426/424/413	**458/453/458**
Logic S4: constant/cumulative/varying domains					
Theorem	197/197/–	220/205/169	111/121/–	**352/338/274**	300/278/245
Non-Thm	001/004/–	004/004/004	036/041/–	082/094/119	**132/146/184**
Solved	198/201/–	224/209/173	147/162/–	**434/432**/393	432/424/**429**

Table 2. No. of monomodal problems in the QMLTP library proved or refuted by HOL-P. The timeout was set to 600s. 60s was given to each subprover of HOL-P; each subprover was applied to both the semantical (sem) and the syntactical (syn) variant.

Logic S4	**constant domains**	**cumulative domains**	**varying domains**
	all (sem/syn)	**all (sem/syn)**	**all (sem/syn)**
Theorem	295 (294/282)	267 (265/256)	241 (238/233)
Non-Theorem	132 (132/132)	146 (146/145)	186 (185/185)
Solved	427 (426/414)	413 (411/401)	427 (423/418)

In terms of theorem proving performance MleanCoP is still the leading system, but its margin of lead over the HOL approach has further decreased (cf. the previous results reported in [1]).

In Table 1 HOL-P has been applied in combination with the semantical encoding of accessibility conditions only. An obvious idea, however, is to test both the semantical and the syntactical encoding. For studying the potential impact of this idea we have conducted further experiments (so far only for S4) in which HOL-P was applied to both versions. Since the overall time limit of 600s per problem was kept, each HOL-P subprover was now given a 60s time limit per problem. Table 2 presents the results of the modified experiment for S4. The first and second numbers in brackets indicate how many problems were solved by the semantical (sem) and the syntactical (syn) approach respectively.

In the Theorem-category the semantical approach performs better. No significant difference can be observed in Non-Theorem-category. The comparison of the overall performance results from Table 2 with those for S4 in Table 1 indicates the following: It makes more sense to run HOL-P in the semantical mode only than to split the time resources and to run HOL-P in both modi (however, what has not been studied yet is the performance of HOL-P when both axiom versions are simply added to one the same problem file).

The individual performances of the subprovers of HOL-P with respect to the experiments in Table 2 are also interesting. They are presented in Table 3.

Table 3. Individual performances of the subprovers of HOL-P in the Theorem-category with respect to the experiments in Table 2. Results are presented for constant domain (const), cumulative domain (cum) and varying domain (vary) semantics.

Logic S4	**Isabelle**	**LEO-II**	**agsyHOL**	**Satallax**
Theorem	const/cum/vary	const/cum/vary	const/cum/vary	const/cum/vary
syn	177/126/120	213/187/163	231/192/171	**244/233/207**
sem	252/215/192	227/203/183	247/206/183	**257/239/214**
total	1082	1176	1230	1394

Table 4. Individual performances of the subprovers of HOL-P in the Non-Theorem-category with respect to the experiments in Table 2.

Logic S4	**Satallax**	**Nitpick**
Non-Theorem	const/cum/vary	const/cum/vary
syn	0/0/0	**132/145/185**
sem	48/56/68	**132/146/185**
total	172	925

Satallax is the strongest prover in the Theorem-category both in the syntactical and the semantical mode. The weak performance of Isabelle in the syntactic mode is surprising, in particular, since Isabelle has performed strong in recent CASC competitions.

In the Non-Theorem-category Nitpick performs significantly stronger than Satallax. The other HOL-P subprovers didn't solve any problems in this category. Interestingly, Nitpick shows nearly equal performance in both the syntactical and the semantical mode, while Satallax solves problems in this category in the semantical mode only.

As a side-result of our experiments we detected some syntax issues in QMLTP problems which were undetected so far: Identifiers for axioms and conjectures were reused; according to TPTP conventions this is not allowed. Examples include 'substitution_of_equivalents', 'reflexivity' and 'transitivity'. These issues were solved manually in the generated `thf0`-files.

7 Summary and Outlook

The `FMLtoHOL` tool enables the application of higher-order automated theorem provers and model finders for solving FML problems encoded in the new `qmf`-syntax. The tool has been evaluated in combination with the higher-order metaprover HOL-P on the QMLTP library. The experiments show that the HOL approach to automate FMLs is very competitive. Regarding the combined performance (no. of proved or refuted problems) the HOL approach performed best.

Future work includes optimizations and extensions of HOL-P and `FMLtoHOL`.

Acknowledgments: We thank Jens Otten, Geoff Sutcliffe, Chad Brown and Jasmin Blanchette for valuable contributions to this work. We also thank the unknown reviewers for several useful comments. The work presented in this paper has been supported by the German Research Foundation DFG under grants BE2501/9-1 and KR858/9-1.

References

1. C. Benzmüller, J. Otten, and T. Raths. Implementing and evaluating provers for first-order modal logics. In *Proc. of ECAI 2012*, Montpellier, France, 2012.
2. C. Benzmüller and L. Paulson. Exploring Properties of Normal Multimodal Logics in Simple Type Theory with LEO-II. In *Reasoning in Simple Type Theory: Festschrift in Honour of Peter B. Andrews*, number 17 in Studies in Logic, Mathematical Logic and Foundations, pages 401–422. College Publications, 2008.
3. C. Benzmüller and L.C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue on Multimodal Logics)*, 7(1):7–20, 2013.
4. C. Benzmüller, F. Theiss, L. Paulson, and A. Fietzke. LEO-II - a cooperative automatic theorem prover for higher-order logic. In *Proc. of IJCAR 2008*, volume 5195 of *LNCS*, pages 162–170. Springer, 2008.
5. J.C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *Proc. of ITP 2010*, volume 6172 of *LNCS*, pages 131–146. Springer, 2010.
6. C.E. Brown. Satallax: An automated higher-order prover. In *Proc. of IJCAR 2012*, volume 7364 of *LNCS*, pages 111 – 117. Springer, 2012.
7. M. Fitting and R.L. Mendelsohn. *First-Order Modal Logic.* Kluwer, 1998.
8. U. Hustadt and R. Schmidt. MSPASS: Modal Reasoning by Translation and First-Order Resolution. In R. Dyckhoff, editor, *Proc. of TABLEAUX 2000*, number 1847 in LNAI, pages 67–71. Springer, 2000.
9. F. Lindblad. agsyHol:. https://github.com/frelindb/agsyHOL, 2012.
10. T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic.* Number 2283 in LNCS. Springer, 2002.
11. J. Otten. leancop 2.0 and ileancop 1.2: High performance lean theorem proving in classical and intuitionistic logic (system descriptions). In *Proc. of IJCAR 2008*, volume 5195 of *LNCS*, pages 283–291. Springer, 2008.
12. J. Otten. Implementing connection calculi for first-order modal logics. In S. Schulz E. Ternovska, K. Korovin, editor, *Intl. WS on the Implementation of Logics*, 2012.
13. T. Raths and J. Otten. The QMLTP problem library for first-order modal logics. In *Proc. of IJCAR 2012*, volume 7364 of *LNCS*, pages 454–461. Springer, 2012.
14. T. Raths, J. Otten, and C. Kreitz. The ILTP Problem Library for Intuitionistic Logic - Release v1.1. *Journal of Automated Reasoning*, 38(1-2):261–271, 2007.
15. G. Sutcliffe. The TPTP problem library and associated infrastructure. *Journal of Automated Reasoning*, 43(4):337–362, 2009.
16. 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.
17. V. Thion, S. Cerrito, and M.C. Mayer. A general theorem prover for quantified modal logics. In *Proc. of TABLEAUX 2002*, volume 2381 of *LNCS*, pages 266–280. Springer, 2002.