EPiC Series in Computing

Volume 46, 2017, Pages 14–30

LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning

Theorem Provers For Every Normal Modal Logic

Tobias Gleißner, Alexander Steen, and Christoph Benzmüller

Freie Universität Berlin, Institute of Computer Science
tobias.gleissner|a.steen|c.benzmueller@fu-berlin.de

Abstract

We present a procedure for algorithmically embedding problems formulated in higher-order modal logic into classical higher-order logic. The procedure was implemented as a stand-alone tool and can be used as a preprocessor for turning TPTP THF-compliant theorem provers into provers for various modal logics. The choice of the concrete modal logic is thereby specified within the problem as a meta-logical statement. This specification format as well as the underlying semantics parameters are discussed, and the implementation and the operation of the tool are outlined.

By combining our tool with one or more THF-compliant theorem provers we accomplish the most widely applicable modal logic theorem prover available to date, i.e. no other available prover covers more variants of propositional and quantified modal logics. Despite this generality, our approach remains competitive, at least for quantified modal logics, as our experiments demonstrate.

1 Introduction

Computer-assisted reasoning in non-classical logics is of increasing interest in artificial intelligence (AI), computer science, mathematics and philosophy. Several powerful automated and interactive theorem proving systems have been developed over the past decades. However, with a few exceptions, most of the available systems focus on classical logics only. In particular for quantified non-classical logics there are only very few systems available to date.

Orthogonal to the development of specialized provers, a shallow semantical embedding approach allows for a quick adaptation of existing higher-order reasoning systems to a broad variety of expressive, non-classical logics [8]. Previous experiments [3, 9] have shown that this approach indeed offers a surprisingly effective automation of the embedded non-classical logics. However, from the users perspective the utilization of the embeddings approach can become rather involved and distracting. Hence, system users may eventually not want to be exposed to the embeddings at all.

The work we present here, which has its roots in a project of the first author [20], bridges between the semantical embedding approach and normal quantified modal logics. Our particular contribution includes:

- A flexible tool that automatically converts higher-order modal logic (HOML) into HOL using a semantical embeddings approach; this obviously includes the conversion of propositional and first-order normal modal logics.

- Previous work [9] realizes such an approach only for first-order modal logics. Moreover, this work has been heavily dependent on the QMLTP library and it required its entire installation. We here extend this work to full HOML, which imposes several additional challenges, and we provide a much leaner and more effective implementation.
- In order to achieve this we need to provide a proper semantics for HOML, which is by no means trivial and which even touches upon some unsettled philosophical issues, for example, regarding higher-order quantification in modal settings; cf. [37, 32]. We here take a pragmatic approach and discuss the range and restrictions of our work accordingly.
- Our tool can be used as a stand-alone preprocessor for problems formulated in the emerging TPTP standard syntax for HOML[1] (which augments the standard TPTP THF syntax for HOL); for preliminary work towards standardized native syntax for HOML see [39, 14].
- When coupled with our tool, standard HOL provers can be used to reason about problems encoded in the native TPTP HOML syntax. We illustrate how this turns HOL reasoners such as Satallax [15] LEO-II [2] and Nitpick [13] into flexible reasoners for HOML.
- We evaluate our system and compare it with the currently best theorem prover for first-order modal logics [6], MleanCoP [29]. For full HOML there are no competitor systems yet available. Hence, we cannot extend our comparative evaluation beyond first-order.
- We contribute many new problems to the TPTP THF library in an extended THF format.

The remainder of this work is structured as follows: §2 briefly introduces the syntax and semantics of higher-order modal logics and discusses its variants. Subsequently, in §3 we sketch the embedding approach. §4 presents the embedding procedure and its implementation. An evaluation of the implementation is given in §5. Finally, §6 concludes and gives an outlook on further developments and improvements.

Further related work. Shallow semantical embeddings into HOL have been studied for various other non-classical logics, including conditional logics [4], hybrid logic [38], intuitionistic logics [7] and more recently, free logics [10] and many-valued logics [33]. All these approaches yield means of automation for the respective non-classical logic. A closely related project aims at an analogous automatic embedding tool for free logic [24], which, in future work, could be merged with the work as presented here to introduce reasoning systems for free higher-order modal logic, which would e.g. be interesting for applications in computational philosophy.

Related approaches at generic theorem proving for various propositional modal logics include the tableau-based theorem systems LoTReC [16], MeTTeL2 [36] and tableau workbench [1]. However, it is unclear whether these approaches scale for quantified modal logics. Also, frameworks for formalizing and reasoning about (modal) proof systems exist that use an encoding to linear logic [27].

2 Higher-Order Modal Logic

We now briefly introduce the syntax and semantics of higher-order multi-modal logics (HOML). HOML can, roughly speaking, be regarded an extension of classical higher-order logic (HOL) [17], augmented with a set of modal operators $\Box^i$, $i \in I$, for some index set I, and equipped

[1] See http://tptp.org/TPTP/Proposals/LogicSpecification.html for the current specification draft.

with an appropriate adaption of Henkin semantics for HOL (cf. [22] and [19]). The presentation of the logic is borrowed from [39] which, in turn, adapts the simplified notation of [26].

2.1 Syntax and Semantics

HOML is a typed logic. The set of *simple types* $\mathcal{T}$ contains all types that are freely generated using the binary function type constructor $\to$ and a set of base types, usually chosen to be $\{o, \iota\}$ for Booleans and individuals, respectively. Terms of HOL are given by the following grammar:

$$s, t ::= c_\tau \mid X_\tau \mid (\lambda X_\tau.\, s_\nu)_{\tau\to\nu} \mid (s_{\tau\to\nu}\, t_\tau)_\nu \mid (\Box^i_{o\to o}\, s_o)_o$$

where $i \in I$, $c_\tau \in \Sigma_\tau$ is a constant symbol from the signature $\Sigma := \bigcup_\tau \Sigma_\tau$ and X_τ is a variable from a countable infinite set $\mathcal{V}_\tau$ of variable symbols for each type $\tau \in \mathcal{T}$. The type of a term is explicitly stated as subscript but may be dropped for legibility reasons if obvious from the context. Terms s_o of type o are *formulas*.

Let I be some non-empty set that serves as an index set for the different modalities. Σ is chosen such that it consists at least of the primitive logical connectives for disjunction, negation, and, for each type, universal quantification and equality. Hence, we have $\{\vee_{o\to o\to o}, \neg_{o\to o}, \Pi^\tau_{(\tau\to o)\to o}, =^\tau_{\tau\to\tau\to o}\} \subseteq \Sigma$ for all $\tau \in \mathcal{T}$. Binder notation $\forall X_\tau.\, s_o$ is used as shorthand for $\Pi^\tau_{(\tau\to o)\to o}\ (\lambda X_\tau.\, s_o)$; and infix notation for the usual binary connectives is employed, e.g. $s \vee t$ instead of $\big((\vee\ s)\ t\big)$. The remaining logical connectives can be defined as usual, e.g by $\wedge_{o\to o\to o} \equiv \lambda s_o.\lambda t_o.\neg(\neg s \vee \neg t)$ and $\Diamond^i_{o\to o} \equiv \lambda s_o.\, \neg\big(\Box^i(\neg s)\big)$. A $\Box^i$-operator's index may be omitted if I is a singleton set.

For the semantics of HOML, the usual concept of a HOL model is augmented with an appropriate notion of Kripke semantics (possible world semantics) [12]. We hereby generalize the notion of models for (first-order) quantified modal logic [21] and adapt it to the full higher-order quantification.

First, we define *frame structures* which collect objects that will be associated with constants, variables and terms of HOML in the following: A *frame* $\mathcal{D} = (D_\tau)_{\tau\in\mathcal{T}}$ is a collection of non-empty sets D_τ, where

(1) D_o is the domain of Booleans with $D_o = \{T, F\}$
where T and F represent truth and falsehood, respectively,

(2) D_i is a non-empty domain of individuals with no further restrictions imposed, and

(3) $D_{\tau\to\nu}$ is a domain of total functions that map D_τ into D_ν.

Next, a *HOML model structure* $\mathcal{M}$ is given by

$$\mathcal{M} = \big(W, \{R^i\}_{i\in I}, \{\mathcal{D}_w\}_{w\in W}, \{\mathcal{I}_w\}_{w\in W}\big)$$

where W is a (non-empty) set of worlds and the $R^i \subseteq W \times W$, $i \in I$, are accessibility relations between these worlds. Additionally, each world w is assigned a frame $\mathcal{D}_w$ and an interpretation function $\mathcal{I}_w$. For any world w, the function $\mathcal{I}_w$ maps a constant symbol $c_\tau \in \Sigma_\tau$ to an appropriate element of $D_\tau \in \mathcal{D}_w$ (the *denotation of c in w*). We assume that the usual logical connectives are given the standard denotation by $\mathcal{I}_w$ in each world $w \in W$, e.g. we have that $\neg_{o\to o}$ denotes the logical negation in each world and analogous for the remaining connectives. Also, we assume $=^\tau_{\tau\to\tau\to o}$ to denote a rigid (world-independent) equality in the following.

A *variable assignment g_w for world w* maps variables $X_\tau \in \mathcal{V}_\tau$, $\tau \in \mathcal{T}$, to elements in D_τ, where $D_\tau \in \mathcal{D}_w$. A variable assignment g is then a collection $g = (g_w)_{w\in W}$. An *X-variant of g*

at world w, written $g[X_\tau/s_\tau]_w$, denotes the variable assignment g' that is identical to g except that in g_w the variable X is mapped to s, i.e. $g'_w(Y) = g_w(Y)$ if $Y \neq X$ and $g'_w(X) = s$.

Finally, the *value* $\|s_\tau\|^{\mathcal{M},g,w}$ of a term s_τ of type τ with respect to a model $\mathcal{M} = \big(W, \{R^i\}_{i\in I}, \{\mathcal{D}_w\}_{w\in W}, \{\mathcal{I}_w\}_{w\in W}\big)$ under variable assignment g at a world $w \in W$ is a value from $D_\tau \in \mathcal{D}_w$ defined by ($i \in I$):

$$\begin{aligned}
\|X_\tau\|^{\mathcal{M},g,w} &= g_w(X)\\
\|c_\tau\|^{\mathcal{M},g,w} &= \mathcal{I}_w(X)\\
\|\,(\lambda X_\tau.\, s_\nu)_{\tau\to\nu}\,\|^{\mathcal{M},g,w} &= f : D_\tau \to D_\nu, \text{with } y \overset{f}{\mapsto} \|s_\nu\|^{\mathcal{M},g[X_\tau/y]_w,w}\\
\|\,(s_{\tau\to\nu}\, t_\tau)_\nu\,\|^{\mathcal{M},g,w} &= \|s_{\tau\to\nu}\|^{\mathcal{M},g,w}\big(\|t_\tau\|^{\mathcal{M},g,w}\big)\\
\|\Box^i_{o\to o}\, s_o\|^{\mathcal{M},g,w} &= \begin{cases} T & \text{if } \|s_o\|^{\mathcal{M},g,v} = T \text{ for all } v \in W \text{ s.t. } (w,v) \in R^i\\ F & \text{otherwise}\end{cases}
\end{aligned}$$

The function $\|.\|$ is well-defined when assuming standard semantics or Henkin semantics [22]. As a result of Gödel's Incompleteness Theorem, complete mechanization of HOML with standard semantics cannot be achieved. Instead, Henkin semantics (also called general semantics) is introduced and assumed for the remainder of this paper. For Henkin semantics, sound and complete calculi exist.

A model structure $\mathcal{M} = \big(W, \{R^i\}_{i\in I}, \{\mathcal{D}_w\}_{w\in W}, \{\mathcal{I}_w\}_{w\in W}\big)$ is called *standard model* if and only if for all $w \in W$ and for all types $\tau, \nu \in \mathcal{T}$ we have that $D_{\tau\to\nu} \in \mathcal{D}_w$ is the complete set of total functions from D_τ to D_ν, i.e. $D_{\tau\to\nu} = D_\nu^{D_\tau}$. Furthermore, the structure $\mathcal{M}$ is called *Henkin model* if and only if for all $w \in W$ and for all types $\tau, \nu \in \mathcal{T}$ the function domain $D_{\tau\to\nu} \in \mathcal{D}_w$ is chosen as a subset $D_{\tau\to\nu} \subseteq D_\nu^{D_\tau}$ of all total functions such that $\|.\|^{\mathcal{M},g,w}$ is total. Every standard model is, of course, also a Henkin model. All references to HOML models in the remainder implicitly refer to Henkin models.

A formula s_o is *valid in model* $\mathcal{M}$, written $\mathcal{M} \models^{\text{HOML}} s_o$, if and only if $\|s_o\|^{\mathcal{M},g,w} = T$ for every variable assignment g and every world $w \in W$. A formula s_o is *valid* (a tautology), denoted $\models^{\text{HOML}} s_o$, if and only if $\mathcal{M} \models^{\text{HOML}} s_o$ for every HOML model $\mathcal{M}$.

2.2 Semantics Variations

The semantics of higher-order modal logic is quite ambiguous. This is due to the existence of various subtle but meaningful variations in some of its facets. Each of those variations have their particular application and are, unfortunately, often assumed implicitly. The most prominent semantics variants are surveyed in the following.

Modality Axiomatizations. The most common variation for a concrete modal logic at hand is the choice of the $\Box^i$-operator's axiomatization. Popular axiom schemes (by no means complete) are displayed in Table 1. It is a well-known fact that certain modal logic formulae correspond to first-order accessibility relation conditions (most notable the so-called Sahlqvist formulae [31]), also displayed in Table 1. Note that there also exist popular relation properties that do not have a modal logic formula equivalent [12], e.g. irreflexivity ($\neg wRw$).

Modal logic systems (denoted by bold-faced names) consist of one or more axiom schemes. As an example, the axiom system **K** only consists of axiom scheme K. More complex systems are then constructed by adding further axiom schemes, e.g. **M** consists of K and M, **S4** consists of K, M and 4, whereas **S5** consists of K, B, M and 5.

Name	Axiom scheme	Condition on r^i	Meta-logical specification of r^i
K	$\Box^i(s \supset t) \supset (\Box^i s \supset \Box^i t)$	—	—
B	$s \supset \Box^i \Diamond^i s$	symmetric	$wR^iv \supset vR^iw$
D	$\Box^i s \supset \Diamond^i s$	serial	$\exists v.wR^iv$
T/M	$\Box^i s \supset s$	reflexive	wR^iw
4	$\Box^i s \supset \Box^i \Box^i s$	transitive	$\big(wR^iv \wedge vR^iu\big) \supset wR^iu$
5	$\Diamond^i s \supset \Box^i \Diamond^i s$	euclidean	$\big(wR^iv \wedge wR^iu\big) \supset vR^iu$
CD	$\Diamond^i s \supset \Box^i s$	functional	$\big(wR^iv \wedge wR^iu\big) \supset v = u$
$\Box$M	$\Box^i(\Box^i s \supset s)$	shift-reflexive	$wR^iv \supset vR^iv$
C4	$\Box^i \Box^i s \supset \Box^i s$	dense	$wR^iv \supset \exists u.wR^iu \wedge uR^iv$
C	$\Diamond^i \Box^i s \supset \Box^i \Diamond^i s$	convergent	$\big(wR^iv \wedge wR^ix\big) \supset \exists u.vR^iu \wedge xR^iu$

Table 1: Popular modal axiom schemes and their corresponding frame condition

There are reasonable applications for every modal system. For example, the modal operator is usually chosen to be **S4** or **S5** when used in an epistemic context. For a multi-modal logic, this choice can be made for every $\Box^i$-operator independently. Additionally, further further bridge rules may be added mediating between the different $\Box^i$-operators.

Quantification. The fairly unrestricted definition of HOML models above yields so-called *varying domains* semantics. Here, we have the situation that denotations $d \in D_\tau \in \mathcal{D}_w$ that exist at a particular world w may not exist in another world v. This is often called the *actualist* interpretation of quantification [21] and states that everything there is (actually) exists, i.e. that there are no merely possible things.

This setting may however not be adequate for all applications of modal logic, in particular in computer science, and is also criticized in the context of metaphysics from so-called *possibilist* positions. The here proposed variant of *constant domain* quantification assumes that the frames of all worlds coincide, i.e. $\mathcal{D}_w = \mathcal{D}_v$ for all worlds $w, v \in W$.

In the setting of *cumulative domains* we still have possibly different frames $\mathcal{D}_w$ for each world w but with the restriction that no denotation object $d \in D_\tau \in \mathcal{D}_w$ may disappear when moving along the accessibility relations. More formally, in cumulative domains we have that if $d \in D_\tau \in \mathcal{D}_w$ then, for all worlds $v \in W$ with $(w, v) \in R^i$, for any $i \in I$, it holds that $d \in D_\tau \in \mathcal{D}_v$. The setting of *decreasing domains* is analogous only that we disallow models that add new denotations to a set D_τ when moving along the accessibility relations.

All of the above variants co-exist and there is still an ongoing dispute about the desired notion of quantification in modal logic [37, 32].

As a further generalization step, one could even extend the idea of domain restriction (constant, varying, cumulative, decreasing) to only apply to certain types and combine them arbitrarily, e.g. that the domain of some type τ is cumulative and all other domains are constant. The philosophical implications of such a setting are cutting-edge and, up to the authors' knowledge, not yet intensively studied.

Rigid and flexible constant. A further dimension of modal logic semantics deals with the dependency of the denotation of constants to the current world: In the above setting of a general HOML model, a constant $c_\tau \in \Sigma_\tau$ may be mapped to different denotations by $\mathcal{I}_w$, depending on the world w at hand (except for the logical connectives such as $\neg$, $\vee$, etc. which are always denoting as usual). We call those symbols *flexible*. However, constant symbols could also be desired to be *rigid*, that is, having the same denotation on all worlds. This can be

acquired by postulating that $\mathcal{I}_w(c) = \mathcal{I}_v(c)$ for all worlds $w \in W$ and all constants c_τ. Aspects of rigidity play an important role for applications in paraconsistent reasoning and when dealing with vagueness.

Finally, it is also possible to assume *some* constants to be flexible and the remaining to be rigid (or vice versa).

Consequence. There is no single meaningful notion of consequence in modal logics. At least two versions of consequence relations have been discussed in the literature [18]: A formula s_o is a *local consequence* of a set of formulas Φ, denoted $\Phi \models^{\text{HOML}}_{\text{local}} s_o$, if and only if for all HOML models $\mathcal{M}$ and for all worlds $w \in W$, $\mathcal{M}, w \models^{\text{HOML}} \phi$ for all $\phi \in \Phi$ implies $\mathcal{M}, w \models^{\text{HOML}} s_o$.

A formula s_o is a *global consequence* of a set of formulas Φ, denoted $\Phi \models^{\text{HOML}}_{\text{global}} s_o$, if and only if for all HOML models $\mathcal{M}$ it holds that, $\mathcal{M}, w \models^{\text{HOML}} \phi$ for all worlds $w \in W$ and $\phi \in \Phi$ implies $\mathcal{M}, w \models^{\text{HOML}} s_o$ for all worlds $w \in W$.

3 Automation via Semantical Embedding

Automation of HOML is realized here using an indirection: The goal is to find equivalent formulations of HOML sentences in classical higher-order logic (HOL). Note that an important aspect of modal logic is that $\Box s$ can be derived if s is a valid formula (this is called necessitation). Nevertheless, $s \supset \Box s$ is in general not a theorem of HOML. In order to capture this non-trivial behavior of the modal operators, the relevant fragments of HOML's Kripke semantics are encoded into HOL. To that end, we first encode all meta-logical ingredients such as connectives as well as validity (in HOML, i.e. $\models^{\text{HOML}}$). Subsequently, we formulate the original modal problem using the encoded meta-logical notions and a translation scheme for HOML terms.

As a result, we can use ordinary theorem proving systems for HOL for reasoning in HOML by inputting the embedded variant of the original problem.

The semantical embedding of modal logics into HOL is intensively discussed in the relevant literature (cf. [8, 11]). We here merely recapitulate the techniques which are later used in §4.

3.1 Classical Higher-Order Logic

Since the target language for the above sketched embedding is HOL we will briefly introduce some notions that are relevant for the remainder of this section.

We assume that in our version of HOL the base types are given by $\{\boldsymbol{o}, \boldsymbol{\iota}, \boldsymbol{\mu}\}$ where, analogously to HOML, $\boldsymbol{o}$ and $\boldsymbol{\iota}$ denote the type of Booleans and individuals, respectively. The new type $\boldsymbol{\mu}$ is later used in the embedding to denote the type of possible worlds. The syntax of HOL is essentially the same as for HOML only that the $\Box^i$ operators are dropped.

We use boldface font for HOL terms and types in order to distinguish them from the ones of HOML, e.g. $\boldsymbol{\lambda X_\tau . s_\nu}$ is a HOL formula of type $\boldsymbol{\tau \to \nu}$.

Models, valuations and validity are defined as usual. We write $\models^{\text{HOL}} \boldsymbol{s}$ to indicate that a formula $\boldsymbol{s_o}$ is (Henkin-) valid in every HOL model. For a thorough introduction of HOL and its semantics, we refer to the literature [5].

3.2 Semantical Embedding

HOML formulas are identified with certain HOL predicates of type $\boldsymbol{\mu \to o}$ where $\boldsymbol{\mu}$ is assumed to denote the type of possible worlds. Intuitively, this allows the evaluation of a formula's truth in a particular world explicitly. The definitions of $\Box^i$ and $\Diamond^i$ are then defined as appropriate

quantifications over the possible worlds (cf. further below). The HOL type $\boldsymbol{\mu \to o}$ is abbreviated as $\boldsymbol{\sigma}$ ("type-lifted Booleans") in the remainder.

Modal axiomatizations. First, for each index $i \in I$ of HOML, we introduce a constant symbol $\boldsymbol{r^i}_{\boldsymbol{\mu\to\mu\to o}}$ to the HOL signature that represents the accessibility relations associated with $\Box^i$. Depending on the desired axiomatization of $\Box^i$, additional restrictions of $\boldsymbol{r_i}$ (cf. Table 1) are postulated employing the correspondence between modal axioms and accessibility relation properties [12].

Logical connectives. Since the type $\boldsymbol{\sigma}$ now serves as the type of (world-dependent) truth values, we need to give definitions to logical connectives operating on them. The encoding $\lceil . \rceil$ of HOML connectives to their HOL equivalents is given by

$$\begin{aligned} \lceil \Box^i_{o\to o} \rceil &= \boldsymbol{\Box^i_{\sigma\to\sigma}} && := \boldsymbol{\lambda S_\sigma.\lambda W_\mu.\forall V_\mu.\ \neg(r^i\ W\ V) \vee S\ V} \\ \lceil \neg_{o\to o} \rceil &= \boldsymbol{\neg_{\sigma\to\sigma}} && := \boldsymbol{\lambda S_\sigma.\lambda W_\mu.\ \neg(S\ W)} \\ \lceil \vee_{o\to o\to o} \rceil &= \boldsymbol{\vee_{\sigma\to\sigma\to\sigma}} && := \boldsymbol{\lambda S_\sigma.\lambda T_\sigma.\lambda W_\mu.\ (S\ W) \vee (T\ W)} \end{aligned}$$

The encoding of universal quantification Π^τ depends on whether this quantification is intended to be using constant domain or varying domain semantics. Let $\boldsymbol{\Pi^{\tau,\mathrm{c}}}$ and $\boldsymbol{\Pi^{\tau,\mathrm{va}}}$ denote the encoding of a constant and varying domain quantification term, respectively, defined by

$$\begin{aligned} \boldsymbol{\Pi^{\tau,\mathrm{c}}_{(\tau\to\sigma)\to\sigma}} &:= \boldsymbol{\lambda P_{\tau\to\sigma}.\lambda W_\mu.\forall X_\tau.\ P\ X\ W} \\ \boldsymbol{\Pi^{\tau,\mathrm{va}}_{(\tau\to\sigma)\to\sigma}} &:= \boldsymbol{\lambda P_{\tau\to\sigma}.\lambda W_\mu.\forall X_\tau.\ \neg(\mathbf{eiw}\ X\ W) \vee (P\ X\ W)} \end{aligned}$$

The definition of constant domain quantification is straight-forward. The encoding of varying domain quantification makes use of the fact that we can simulate varying domains by postulating that the single frame $\mathcal{D}$ of HOL consists over the union of all $\mathcal{D}_w$ of HOML and the predicate $\boldsymbol{\mathbf{eiw}_{\tau\to\sigma}}$ controls/specifies whether an object of type τ indeed "exists" in the given world. Such a predicate is needed for all types. The remaining idea of the encoding of universal quantification is that a given property of objects, in order to be a universal property, needs only to hold for those objects that indeed "exist", i.e. , **eiw** is used as a guard. In order to postulate cumulative or decreasing domains, an according property on **eiw** is added as axiom.

Embedding the problem. For each type τ occurring in the HOML problem we define the embedding (or type-lifting) $\lceil \tau \rceil$ of a type τ by

$$\lceil \tau \to \nu \rceil = \lceil \tau \rceil \to \lceil \nu \rceil \qquad \text{with } \lceil o \rceil = \boldsymbol{\sigma} := \boldsymbol{\mu \to o}$$

The type-lifting of HOML type ι now depends on whether we assume rigid constants or flexible constants. In the first case we set $\lceil \iota \rceil = \boldsymbol{\iota}$, in the latter case $\lceil \iota \rceil = \boldsymbol{\mu \to \iota}$.[2] We extend the definition of $\lceil . \rceil$ to HOML terms by

$$\lceil c_\tau \rceil = \boldsymbol{c_{\lceil\tau\rceil}} \qquad \lceil X_\tau \rceil = \boldsymbol{X_{\lceil\tau\rceil}} \qquad \lceil \lambda X_\tau.\, s_\nu \rceil = \boldsymbol{\lambda}\lceil X_\tau \rceil.\, \lceil s_\nu \rceil \qquad \lceil s_{\tau\to\nu}\ t_\tau \rceil = \lceil s_{\tau\to\nu} \rceil\ \lceil t_\tau \rceil$$

Hence, all the constant symbols and the variables are lifted to (bold) equivalents in HOL.

Finally, we encode the notion of HOML validity resp. consequence with aid of two further meta-logical definitions $\lfloor . \rfloor$ and $\mathcal{A}(.)$. Both are grounding terms of type $\boldsymbol{\sigma}$ (i.e. formulas of

[2] We fix the type-lifting of functional types to be rigid. There are applications in metaphysics where flexible

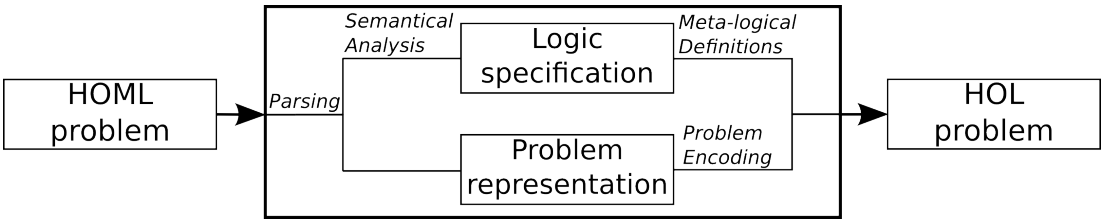

Figure 1: Bird's eye perspective of the automated embedding process

HOML) to type o (formulas of HOL) and assert that the respective formula is valid. The difference between these two notions is that the first one encodes global consequence semantics while the latter can be used to encode local consequence.

When assuming global consequence semantics, we simply ground all terms s_σ by

$$\lfloor s_\sigma \rfloor := \forall W_\mu . s\ W$$

Here, the quantification over all worlds is associated with the term itself, corresponding to the definition of $\models^{\text{HOML}}_{\text{global}}$. For local consequence, the quantification over all worlds needs to be at the outermost scope, collecting all axioms within its range. This can be done by grounding all terms s_σ using

$$\mathcal{A}(s_\sigma) := s\ w_{\mathbf{actual}}$$

where $w_{\mathbf{actual}}$ is a uninterpreted constant symbol of type μ denoting an (arbitrary but fixed) actual world for simulating the universal quantification over worlds corresponding to $\models^{\text{HOML}}_{\text{local}}$.

Finally, we have that

$$\begin{aligned} \Phi \models^{\text{HOML}}_{\text{global}} s_o \quad &\text{if and only if} \quad \{\lfloor \lceil \phi_o \rceil \rfloor \mid \phi_o \in \Phi\} \models^{\text{HOL}} \lfloor \lceil s_o \rceil \rfloor \\ \Phi \models^{\text{HOML}}_{\text{local}} s_o \quad &\text{if and only if} \quad \{\mathcal{A}(\lceil \phi_o \rceil) \mid \phi_o \in \Phi\} \models^{\text{HOL}} \mathcal{A}(\lceil s_o \rceil) \end{aligned}$$

A proof for global consequence and constant domain semantics was presented in [8]. Proofs for the remaining semantical variants are currently pursued.

Using the above approach, it is also possible to explore are more general field of normal modal logics by flexibly postulating Kripke frame restrictions that are not expressible as modal logic formulae or even first-order condition. For example, it is possible to restrict an accessibility relation r^i to be irreflexive while there exists to corresponding modal logic formula. Note that, however, higer-order frame restrictions are still evaluated under Henkin models only.

4 Automated embedding

The goal of this work is to extend ordinary HOL ATPs to function as specialized reasoning tools for higher-order quantified modal logic with as less technical overhead for the user as possible. This implies that most of the involved embedding and type-lifting of formulas should be hidden and also that only minor modifications are imposed on the ordinary ATP usage workflow.

For an automated embedding procedure, both the source logic's semantics as well as the original problem statement needs to be translated to adequate HOL terms. Since one and

functions are required for properly formulating certain non-trivial arguments. This can be allowed by defining $\lceil \tau \to \nu \rceil = \mu \to \lceil \tau \rceil \to \lceil \nu \rceil$. However, in this paper we restrict the embedding to use rigid function types only.

the same problem statement may be studied in various semantical contexts and it cannot automatically be inferred which semantics is intended, the information on the concrete desired logic needs to be provided by the user. To that end, the standard input syntax from HOL ATP is extended to allow the inclusion of an appropriate logic specification to the problem file. Hence, it is transparent to the user in what semantical context the following problem is formulated while leaving the concrete syntax of the problem's formulas and definitions unchanged.

The ATP usage workflow is more or less unchanged since the problem encoding can be done automatically in a preprocessing step. A complete schematic invocation of this preprocessing step is displayed at Fig. 1. The input of this process is a problem statement formulated in a well-defined syntax for HOML as described further below. After parsing, the meta-logical contents of the problem statement (the semantical specification) are processed. This produces HOL encodings of corresponding meta-logical notions of HOML. In a second step, the problem itself is embedded as described in §3.2. Both parts are concatenated and written out as result.

Note that the result of the embedding procedure is a plain TPTP THF problem [35]. This means that any TPTP-compliant HOL ATP can then be given the output of the embedding procedure without any adjustment to the system. The only slight modification to the ATP usage workflow is that the above preprocessing tool is prepended to the problem processing pipeline.

Logic specification. A de-facto standard representation of HOL problems for automated theorem provers (ATPs) is given by the TPTP THF dialect [35]. This representation syntax is supported by most current HOL reasoners, including Satallax [15], LEO-II [2], Isabelle/HOL [28], Nitpick [13] and many others.

As depicted further above, the syntax of HOML is a conservative extension of the standard HOL syntax. Hence THF can easily be augmented by introducing the modal operators `$box` and `$dia` as primitive connectives (for a mono-modal settings). For example, the HOML formula $\forall X_\iota.\Box(p\ X)$ then corresponds to `! [X:$i] : ( $box @ ( p @ X ) )`. In a multi-modal environment, the new primitives `$box_int` and `$dia_int` can be applied to an integer which serves as the index of the modal operator. The HOML formula $\forall X_\iota.\Box^1(p\ X)$ hence corresponds to `! [X:$i] : ( $box_int @ 1 @ ( p @ X ) )`.

The desired semantics of a problem's underlying modal logic is specified within the problem statement itself using a meta-logical declaration statement (using a formula annotated with the role `logic`). Such an approach was fostered in earlier work [39] and subsequently improved and enhanced to yield a TPTP standard proposal that is still under development. The `logic` statement specifies the concrete modal logic using a `$modal` meta-variable which is assigned the semantics for each semantic dimension (cf. §2.2) as follows

```
thf(simple_s5, logic, ($modal := [
   $constants := $rigid,
   $quantification := $constant,
   $consequence := $global,
   $modalities := $modal_system_S5 ])).
```

Here, as an example, an **S5** modal logic is postulated with rigid constants, constant domain semantics and global consequence.

The specification of semantical properties is extended to individual objects such that a dedicated semantics can be given for each type, constant, axiom or modal operator (depending on the semantic dimension at hand) separately. For example, quantification over some type *human* can be postulated to employ varying domains whereas all other quantifications remain constant domain. Analogous choices can be made for rigidity (per constant symbol), conse-

quence (per axiom or conjecture) and axiomatization (per modal operator). A more involved logic specification reads

```
thf( mydomain_type , type , ( human : $tType ) ).
thf( myconstant_declaration , type , ( myconstant : $i ) ).
thf( complicated_s5 , logic , ( $modal := [
    $constants := [ $rigid , myconstant := $flexible ] ,
    $quantification := [ $constant , human := $varying ] ,
    $consequence := [ $global , myaxiom := $local ] ,
    $modalities := [ $modal_system_S5 , $box_int @ 1 := $modal_system_T ] ] ) ).
```

As shown, object-specific declarations are specified by assigning a list starting with the default value of a property followed by assignments of objects to values. The same applies to different modal operators in multi-modal settings. The current list of supported values includes

- `$constants` $\in \{$`$rigid`, `$flexible`$\}$
- `$quantification` $\in \{$`$varying`, `$cumulative`, `$decreasing`, `$constant`$\}$
- `$consequence` $\in \{$`$global`, `$local`$\}$
- `$modalities` Use `$modal_system_X` for `X` $\in \{$`K`, `D`, `T`, `S4`, `S5`, ...$\}$[3]

Embedding procedure. The algorithm for encoding the HOML problem into HOL implements the steps roughly sketched in §3.2. First, meta-logical definitions are encoded into THF:

```
thf(w, type, ( w_type:$tType ) ). %% The world type
thf(r, type, (r:w_type>w_type>$o) ). %% One accessibility relation
```

The type `w_type` of possible worlds corresponds to $\boldsymbol{\mu}$ of §3.2. The optional constraints on `r` are imposed by simply applying the according properties to it as an axiom.

```
thf(mrefl_type, type, ( mrefl : (w_type>w_type>$o)>$o ) ).
thf(mrefl_def, definition, ( mrefl = (^ [R:w_type>w_type>$o] : ![A:w_type]: (R@A@A)))).
thf(r_mrefl, axiom, ( mrefl @ r ) ).
```

Next, the grounding operators $\lfloor . \rfloor$ and $\boldsymbol{\mathcal{A}}$ are defined and appended to the output:

```
thf(mvalid_type, type, ( mvalid: (w_type>$o)>$o ) ).
thf(mvalid, definition, ( mvalid = (^ [S:w_type>$o] : ! [W:w_type] : (S@W)))).
thf(mcurworld_type, type , ( mcurworld: w_type ) ).
thf(mactual_type, type, ( mactual: ( ( w_type>$o ) >$o ) ) ).
thf(mactual, definition, ( mactual = ( ^ [Phi:(w_type>$o)] : ( Phi @ mcurworld ) ) ) ).
```

Subsequently, the embedded variants of the logical connectives are defined (type declarations omitted). They will later be replace the native connectives of the HOML problem statement.

```
thf(mnot, definition, ( mnot = (^ [A:w_type>$o,W:w_type] : ~(A@W)))).
thf(mor, definition, ( mor = (^ [A:w_type>$o,B:w_type>$o,W:w_type]: ((A@W)|(B@W))))).
thf(mbox, definition, ( mbox = (^ [A:w_type>$o,W:w_type]:
                                  ! [V:w_type] : ( (rel_r@W@V) => (A@V) ) ))).
```

Since there are hardly any HOL ATP systems that robustly support the polymorphic HOL syntax enhancement TH1, we need to explicitly define and include quantifiers for all types that were quantified over in the original problem. Consider the following example:

[3]One can also define the axiomatization of a modal operator using the modal axioms schemes explicitly instead of the system names, e.g. `$box := [$modal_axiom_K, $modal_axiom_B, ...]`.

```
thf(mforall_const_i_ty, type, ( mforall_const_i : (($i>w_type > $o)>w_type>$o) ) ).
thf(mforall_const_i, definition, ( mforall_const_i =
                    ( ^ [A:($i)>w_type>$o,W:w_type] : ! [X:($i)] : (A @ X @ W)) )).
thf(eiw_type_i, type, ( eiw_i : ($i>w_type > $o) )).
thf(eiw_nonempty_i, axiom, (! [W:w_type]: ( ? [X:($i)] : (eiw_i @ X @ W)))).
thf(mforall_vary_i_ty, type, ( mforall_vary_i : (($i>w_type>$o)>w_type>$o) )).
thf(mforall_vary_i, definition, ( mforall_vary_i =
       (^[A:($i)>w_type>$o,W:w_type]: ![X:($i)]: ((eiw_i @ X @ W) => (A @ X @ W))) )).
```

The first two declarations define constant domains quantification for type `i`. Recall that varying domains quantification requires a predicate, denoted `eiw` for *exists in world*, as a guard to nested quantification. Such an `eiw` predicate is, again, needed for each type so that we define the predicate and its non-emptiness restriction first, and then output the definition of varying domain quantification (of type `i` on lines 4 and 5 of the above example). Additional restrictions to the `eiw` predicate are subsequently included, e.g. when using cumulative domain semantics:

```
thf(eiw_cumul_i, axiom, (! [W:w_type,V:w_type,C:$i]:
                           ((r @ W @ V) => ((eiw_i @ C @ W) => (eiw_i @ C @ V))) )).
```

Note that also for each user-constant `c` of type `t` of the input problem, an assertion of the existence of `c` in all worlds is needed. If a respective `eiw` predicate has not been introduced yet, it is introduced now (omitted here).

```
thf(c_eiw, axiom, (! [W:w_type]: (eiw_t @ c @ W))).
```

However, this axiom is included at the end of the resulting THF statement in order to respect symbol occurrence restrictions (a symbol needs to be introduced before its first usage in a formula or term).

This concludes the introduction of new types, definitions and axioms. Next, the actual input problem is transformed. The first step is to lift all occurring types according to the embeddings rules. As an example, every occurrence of type `$o` in the HOML problem is transformed to `w_type>$o`. Then, every connective is replaced by its embedded counterpart. As an example, `( a & ~(b) )` is transformed to `( mand @ a @ ( mnot @ b ) )`. The only exception is the replacement of quantifiers since the embedded equivalents are combinators rather than binding mechanisms as used in the original problem statement. Hence a λ-abstraction for the variable name to be quantified is inserted on top on the replacement. As an example, `( ![X:$i]: (p @ X) )` is transformed to `( mforall_const_i @ (^[X:$i]: (p @ X)) )` when assuming constant domain semantics or to `( mforall_vary_i @ (^[X:$i]: (p @ X)) )` when assuming varying domain semantics.

The final step is to ground all formulas (i.e. terms of type `w_type>$o` after lifting) to type `$o`. To that end, the actuality operator is applied to all statements labeled with local consequence and the validity operator to those labeled with global consequence, yielding `( mactual @ a )` or `( mvalid @ a )`, respectively, for some formula `a`.

The tool. The above procedure was implemented as an open-source stand-alone tool written in Java 8 and is freely available online[4]. The tool reads an input problem file formulated in the above described augmented modal THF representation and writes the embedded problem to a new file. It is invoked by

```
./embed -f modal -i <input file> -o <result file>
```

[4]See https://github.com/TobiasGleissner/embed_modal.

Table 2: Results of validity checking of (BF), (CBF), (BF) + (CBF) and (NE).

	(BF)				**(CBF)**				**(BF) + (CBF)**				**(NE)**			
	vary	cumul	decr	const	vary	cumul	decr	const	vary	cumul	decr	const	vary	cumul	decr	const
Valid	X	X	✓	✓	X	✓	X	✓	X	X	X	✓	X	✓	X	✓
LEO-II	$< 1s$	†	$< 1s$	$< 1s$	$< 1s$	$< 1s$	†	$< 1s$	$< 1s$	†	†	$< 1s$	$< 1s$	$< 1s$	†	$< 1s$
Satallax	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$
Nitpick	$10s$	$8s$	†	†	$10s$	†	$10s$	†	$10s$	$8s$	$11s$	†	$7s$	†	$7s$	†
MLeanCoP*	$< 1s$	$< 1s$	‡	$< 1s$	$< 1s$	$< 1s$	‡	$< 1s$	$< 1s$	$< 1s$	‡	$< 1s$	———		=	———

*MleanCoP does not support scheme K, instead D was employed. A first-order encoding was used. †: Timeout/GaveUp ‡: Semantics not supported =: Equality not supported.

If the input problem does not contain a semantical specification the tool can externally provide different semantical settings using a command-line option `-semantics <semantics>`. The tool can recursively embed all files of a directory if the `<input file>` is not a file but a directory.

5 Evaluation

In this section, the embedding tool is evaluated using two distinct applications: Firstly, we demonstrate how the embedding tool can be used to study relevant philosophical discussions that require analysis with respect to multiple semantical settings.

Secondly, the correctness and performance of the embedding tool is evaluated using the complete set of mono-modal problems from the QMLTP problem library [30].

For the measurement, Satallax 2.7 [15] and LEO-II 1.7.0. [2] (compiled under Debian Linux 3.16.0 using OCaml 4.02.3) were used. Additionally, Nitpick [13] from Isabelle's 2016 distribution is used as counter model finder for cross verification. One sophisticated representative of native modal logics provers, MleanCoP [29], is used for performance comparison. For comparability reasons, rigid constants are assumed in this section.

Actualism vs. Possibilism. One of the most prominent controversial consequence of the possibilists's interpretation of modal logic semantics (i.e. modal quantification semantics) is the validity of all instances of the Barcan Formula (BF), the Converse Barcan Formula (CBF) and Necessary Existence (NE) [25]. For type ι, the instances are given by

$$\text{(BF)}\quad \forall X_\iota.\Box pX \supset \Box\forall X_\iota.pX \qquad \text{(CBF)}\quad \Box\forall X_\iota.pX \supset \forall X_\iota.\Box pX \qquad \text{(NE)}\quad \forall X_\iota.\Box\exists Y_\iota.Y = X$$

where $p_{\iota\to o}$ is some predicate. As an example, (BF) has vast implications on so-called *Possibilia*, i.e. that the mere possibility of existing objects of some property implies the actual existence of those object having possibly that property. Similar arguments exist for (CBF) and (NE) [25].

In order to study the validity of the above instances of (BF) and (CBF), the conjunction of both, denoted (BF) + (CBF) in the following, and (NE) under different semantical settings, we represent the formulas in modal THF syntax. Subsequently, the embedding procedure is invoked for each of the four problems and each quantification semantics individually.

Table 2 summarizes the results (30s time limit). The first line indicates the actual validity of the respective formula, where valid instances are marked with ✓ and non-valid (counter satisfiable) instances with X. The remaining lines indicate the results (time taken for solving) from the corresponding reasoning systems. LEO-II, Satallax and MleanCoP are able to correctly prove and disprove (almost) every instance of (BF), (CBF) and (BF) + (CBF) in less than one second. For all counter-satisfiable instances, Nitpick can give a comprehensive, finite counter model in small time. Similar results apply to (NE) using the embedding approach. MleanCoP cannot be applied here since it does not support native equality.

Table 3: Number of solved problems under local consequence (‡: Semantics not supported)

Semantics	Reasoning system															
	LEO-II				Nitpick				Satallax				All	MleanCoP		
	Σ	THM	CSA	CSA*	Σ	THM	CSA	CSA*	Σ	THM	CSA	CSA*	U	Σ	THM	CSA
K/vary	81	81	0	65	0	0	0	286	105	105	0	135	108	‡		
K/cumul	98	98	0	29	0	0	0	267	122	122	0	122	125	‡		
K/decr	94	94	0	29	0	0	0	264	118	118	0	124	122	‡		
K/const	216	146	70	-	237	0	237	-	284	167	117	-	427	‡		
D/vary	133	90	43	1	205	0	205	68	123	114	9	0	1	454	184	270
D/cumul	124	107	17	0	185	0	185	67	134	133	1	0	0	451	206	245
D/decr	104	104	0	16	0	0	0	251	130	130	0	1	133	‡		
D/const	211	160	51	-	223	0	223	-	192	181	11	-	58	441	223	218
T/vary	169	131	38	2	130	0	130	95	239	164	75	29	5	377	221	156
T/cumul	160	149	11	0	108	0	108	92	248	186	62	28	4	381	250	131
T/decr	141	141	0	11	0	0	0	222	177	177	0	91	180	‡		
T/const	247	204	43	-	142	0	142	-	317	234	83	-	54	382	270	112
S4/vary	155	155	0	0	110	0	110	101	270	200	70	28	2	411	286	125
S4/cumul	179	179	0	0	86	0	86	95	286	227	59	27	3	443	348	95
S4/decr	165	165	0	0	0	0	0	190	217	217	0	86	220	‡		
S4/const	237	237	0	-	116	0	116	-	358	283	75	-	47	445	364	81
S5/vary	187	187	0	0	81	0	81	86	290	237	53	28	1	450	358	92
S5/cumul	230	230	0	0	35	0	35	100	312	284	28	25	3	475	436	39
S5/decr	230	230	0	0	0	0	0	119	284	284	0	53	293	‡		
S5/const	275	275	0	-	70	0	70	-	375	321	54	-	38	475	436	39

QMLTP. A more extensive evaluation of the previously presented tool is presented in the following. As a first step, the 580 mono-modal problems of the QMLTP library [30] (version 1.1) were translated into modal THF in order to the applicable to our approach. As it turns out, some of the QMLTP problems are not stated in proper extended TPTP FOF syntax (mostly missing parentheses); these problems have been adjusted by hand. Then, the problems were embedded into plain THF for modal systems **K**, **D**, **T**, **S4**, **S5** (these are supported by QMLTP), all quantification semantics and both consequences (local and global). While our system is by no means restricted to these modal systems, we do restrict the evaluation to those since there are no other theorem provers available for comparison for further semantical variations. Each previously mentioned reasoning system was tested using a time limit of $60s$ and ran on an eight core system (2x AMD Opteron 2376 Quad Core) with 32 GB RAM.

Table 3 and Table 4 present the number of solved problems for each prover and each semantical setting in local and global consequence, respectively. The columns Σ, THM and CSA show the number of solved problems and the number of (thereof) proved and refuted problems, respectively. The number of problems uniquely solved by the embedding approach (i.e. any HOL ATP but not by MleanCoP) is denoted U.

The post-processing of the benchmark results involved an ATP-to-ATP crosscheck for excluding contradicting system results. Similarly, an ATP-QMLTP crosscheck was conducted (where possible) for the local consequence setting for comparing the results of the embedding to the status in the QMLTP header file. The main observations of this checks include

- No ATP-to-ATP discrepancies were found among the HOL ATP.
- Two ATP-to-ATP/ATP-QMLTP soundness mismatches (modulo semantics) were found for problems SYM052+1 and SYM056+1 (Embedding: Theorem, MleanCoP and QMLTP: CSA), resulting in nine mismatches counting different semantics. After further investigation and communication with Jens Otten, it is apparent that this results from a lapse in the original problem statement. The QMLTP defines its own equality due to complications involving notions of modal equality and therefore the (in-)equality sign "=" ("!=") is assumed an uninterpreted symbol and should not occur in the QMLTP. Nevertheless, SYM052+1 contains such an inequality sign. Consequently, MleanCoP refutes

Table 4: Number of solved problems under global consequence

Semantics	Reasoning system									
	LEO-II			Nitpick			Satallax			all
	Σ	THM	CSA$^{(*)}$	Σ	THM	CSA$^{(*)}$	Σ	THM	CSA$^{(*)}$	Σ
K/vary	85	85	67	0	0	380	117	117	100	119
K/cumul	102	102	31	0	0	350	135	135	87	137
K/decr	97	97	30	0	0	356	130	130	89	132
K/const	231	152	79	256	0	256	266	184	82	453
D/vary	91	91	44	0	0	372	126	126	8	128
D/cumul	111	111	17	0	0	338	147	147	1	149
D/decr	104	104	17	0	0	344	143	143	1	145
D/const	214	164	50	240	0	240	208	198	10	451
T/vary	131	131	39	0	0	308	172	172	76	175
T/cumul	149	149	11	0	0	285	195	195	62	198
T/decr	140	140	10	0	0	296	186	186	63	189
T/const	248	209	39	161	0	161	304	249	55	423
S4/vary	157	157	0	0	0	282	199	199	70	203
S4/cumul	181	181	0	0	0	244	227	227	58	234
S4/decr	169	169	0	0	0	261	217	217	58	221
S4/const	240	240	0	122	0	122	334	287	47	423
S5/vary	188	188	0	0	0	233	238	238	53	243
S5/cumul	228	228	0	0	0	180	285	285	25	292
S5/decr	230	230	0	0	0	181	286	286	25	293
S5/const	273	273	0	72	0	72	351	325	26	415

Due to lack of comparison results, only the CSA values for constant domain semantics are counted into Σ.

the problem while the HOL ATPs do interpret the embedded equality sign.

- Apart from the two problems above, no soundness issues were found.
- Some problems where deemed CSA by the HOL ATP while being a theorem according to QMLTP and MleanCoP. A study of finite counter models produced by Nitpick show that the embedding allows a more general notion of modal model structures when considering non-constant quantification semantics that are not captured by native modal logic provers. One example is that, in the embedding, we can construct models where there is an object that does not exist in any world. Such a construction is highly interesting from a metaphysical point of view. One possibility to restrict models to only range over "ordinary" structures is to simply forbid this situation by additional axioms. It is nevertheless not clear whether this suffices. Those extra CSA results are displayed as CSA* in the table.
- 183 results were contributed by the embedding approach (refutations counted only for constant domain semantics) for the semantical settings supported by the QMLTP and which were previously unsolved by any QMLTP prover (including MleanCoP).

The remaining measurement results indicate that our approach is indeed competitive compared to native modal logic provers. Also, in contrast to the existing modal logic provers, the semantical parameters can be adjusted more flexibly in the embedding approach, i.e. 5971 results were contributed for semantical settings not supported by QMLTP. The number of CSA results is not as high as possible since we cannot safely distinguish between CSA results and CSA* results at the moment and hence only count those confirmed by MleanCoP onto CSA.

6 Conclusion and Further work

In this work, we presented means for encoding various versions of higher-order modal logic into classical HOL. A syntax for modal THF problems was presented and an automated procedure

for embedding problems formulated in such a syntax was outlined and implemented as a standalone tool in Java. It can be used as a preprocessor for turning common HOL reasoning systems into reasoners for almost every normal modal logic.

The effectiveness of such an embedding approach was studied using relevant formulas from philosophy and on the bases of the QMLTP library. To that end, all 580 mono-modal problems of the QMLTP were translated into modal THF[5]. The evaluation results indicate that the presented approach is indeed competitive and supports more semantical variants than any modal logic prover. As further results, some apparent incongruities in the QMLTP were discovered. Also, the availability of a more general notion of model structures for modal logic was found that can be utilized using our embedding approach. Those model structure should be investigated further since they may be suitable for certain philosophical applications. Additional axiomatizations for achieving completeness for non-constant domains with respect to the usual model structures were sketched but need to be studied further. After such an adjustment, the embedding tool should yield even more competitive results.

The inclusion of the here presented tool into the Leo-III prover [34] remains future work but is straight-forward. Also, a leaner HOL encoding of the HOML problems can be achieved by offering translations to polymorphic HOL, i.e. into TH1 syntax [23].

Acknowledgments

This work has been supported by the German National Research Foundation (DFG) under grant BE 2501/11-1 (Leo-III). We thank Geoff Sutcliffe for his advice and cooperation on the extension of the THF syntax, Jens Otten for his insightful comments on the evaluation and Harold Boley for his comments on earlier versions of this work. Furthermore, we thank the anonymous reviewers for their helpful improvements.

References

[1] Abate, P., Gor, R.: The Tableau Workbench. Electronic Notes in Theoretical Computer Science 231, 55 – 67 (2009)

[2] Benzmüller, C., Paulson, L.C., Sultana, N., Theiß, F.: The higher-order prover LEO-II. Journal of Automated Reasoning 55(4), 389–404 (2015)

[3] Benzmüller, C.: HOL provers for first-order modal logics — experiments. In: Benzmüller, C., Otten, J. (eds.) ARQNL 2014. Automated Reasoning in Quantified Non-Classical Logics. EPiC Series in Computing, vol. 33, pp. 37–41. EasyChair (2015)

[4] Benzmüller, C.: Cut-Elimination for Quantified Conditional Logic. J. of Philosophical Logic (2016)

[5] Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. Journal of Symbolic Logic 69(4), 1027–1088 (2004)

[6] Benzmüller, C., Otten, J., Raths, T.: Implementing and Evaluating Provers for First-order Modal Logics. In: Raedt, L.D., et al. (eds.) ECAI 2012. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 163–168. IOS Press, Montpellier, France (2012)

[7] Benzmüller, C., Paulson, L.: Multimodal and intuitionistic logics in simple type theory. The Logic Journal of the IGPL 18(6), 881–892 (2010)

[8] Benzmüller, C., Paulson, L.: Quantified Multimodal Logics in Simple Type Theory. Logica Universalis (Special Issue on Multimodal Logics) 7(1), 7–20 (2013)

[5]The modal THF problems are available at https://github.com/TobiasGleissner/QMLTP.

[9] Benzmüller, C., Raths, T.: HOL based first-order modal logic provers. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) Proc. Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR). LNCS, vol. 8312, pp. 127–136. Springer, Stellenbosch, South Africa (2013)

[10] Benzmüller, C., Scott, D.: Automating free logic in Isabelle/HOL. In: Greuel, G.M., Koch, T., Paule, P., Sommese, A. (eds.) Mathematical Software – ICMS 2016, 5th International Congress, Proceedings. LNCS, vol. 9725, pp. 43–50. Springer, Berlin, Germany (2016)

[11] Benzmüller, C., Woltzenlogel Paleo, B.: Higher-order modal logics: Automation and applications. In: Paschke, A., Faber, W. (eds.) Reasoning Web 2015. pp. 32–74. No. 9203 in LNCS, Springer, Berlin, Germany (2015), (Invited paper)

[12] Blackburn, P., van Benthem, J.F., Wolter, F.: Handbook of modal logic, vol. 3. Elsevier (2006)

[13] Blanchette, J., Nipkow, T.: Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In: Proc. of ITP 2010. LNCS, vol. 6172, pp. 131–146. Springer (2010)

[14] Boley, H., Benzmüller, C., Luan, M., Sha, Z.: Translating higher-order modal logic from RuleML to TPTP. In: Giurca, A., et al. (eds.) Proceedings of the RuleML 2016 Challenge, the Special Industry Track and the RuleML 2016 Doctoral Consortium hosted by the 10th International Web Rule Symposium (RuleML 2016). CEUR Workshop Proceedings, vol. 1620. CEUR-WS.org (2016)

[15] Brown, C.: Satallax: An automated higher-order prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) Proc. of IJCAR 2012. LNAI, vol. 7364, pp. 111 – 117. Springer (2012)

[16] del Cerro, L.F., et al.: Lotrec : The Generic Tableau Prover for Modal and Description Logics. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) Automated Reasoning, First Int. Joint Conf., IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings. LNCS, vol. 2083, pp. 453–458. Springer (2001)

[17] Church, A.: A formulation of the simple theory of types. Journal of Symbolic Logic 5, 56–68 (1940)

[18] Fitting, M., Mendelsohn, R.: First-Order Modal Logic. Synthese Library Studies in Epistemology Logic, Methodology, and Philosophy of Science Volume 277, Springer (1998)

[19] Gallin, D.: Intensional and higher-order modal logic (1975)

[20] Gleißner, T.: Converting Higher-Order Modal Logic Problems into Classical Higher-Order Logic. Bsc. thesis, Freie Universität Berlin, Institute of Computer Science, Berlin, Germany (2016)

[21] Goldblatt, R.: Quantifiers, Propositions and Identity: Admissible Semantics for Quantified Modal and Substructural Logics. Lecture Notes in Logic, Cambridge University Press (2011)

[22] Henkin, L.: Completeness in the theory of types. Journal Symbolic Logic 15(2), 81–91 (1950)

[23] Kaliszyk, C., Sutcliffe, G., Rabe, F.: TH1: the TPTP typed higher-order form with rank-1 polymorphism. In: Fontaine, P., Schulz, S., Urban, J. (eds.) Proceedings of the 5th Workshop on Practical Aspects of Automated Reasoning. CEUR Workshop Proceedings, vol. 1635, pp. 41–55. CEUR-WS.org (2016)

[24] Makarenko, I.: Automatisierung von freier Logik in Logik höherer Stufe. Bsc. thesis, Freie Universität Berlin, Institute of Computer Science, Berlin, Germany (2016)

[25] Menzel, C.: Actualism. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, winter 2016 edn. (2016)

[26] Muskens, R.: Higher order modal logic. Handbook of modal logic 3 (2007)

[27] Nigam, V., Pimentel, E., Reis, G.: An extended framework for specifying and reasoning about proof systems. J. Log. Comput. 26(2), 539–576 (2016)

[28] Nipkow, T., Paulson, L., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic. No. 2283 in LNCS, Springer (2002)

[29] Otten, J.: Mleancop: A connection prover for first-order modal logic. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Vienna, Austria, July 19-22, 2014. Proceedings. LNCS, vol. 8562, pp. 269–276. Springer (2014)

[30] Raths, T., Otten, J.: The QMLTP Problem Library for First-Order Modal Logics. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS, vol. 7364, pp. 454–461. Springer (2012)

[31] Sahlqvist, H.: Completeness and correspondence in the first and second order semantics for modal

logic. Studies in Logic and the Foundations of Mathematics 82, 110–143 (1975)

[32] Stalnaker, R.: Mere Possibilities: Metaphysical Foundations of Modal Semantics. Carl G. Hempel Lecture Series, Princeton University Press (2012)

[33] Steen, A., Benzmüller, C.: Sweet SIXTEEN: Automation via embedding into classical higher-order logic. Logic and Logical Philosophy 25, 535–554 (2016)

[34] Steen, A., Wisniewski, M., Benzmüller, C.: Agent-based HOL reasoning. In: Greuel, G.M., Koch, T., Paule, P., Sommese, A. (eds.) Mathematical Software – ICMS 2016, 5th International Congress, Proceedings. LNCS, vol. 9725, pp. 75–81. Springer, Berlin, Germany (2016)

[35] Sutcliffe, G., Benzmüller, C.: Automated reasoning in higher-order logic using the TPTP THF infrastructure. Journal of Formalized Reasoning 3(1), 1–27 (2010)

[36] Tishkovsky, D., Schmidt, R.A., Khodadadi, M.: The tableau prover generator MetTeL2. In: European Workshop on Logics in Artificial Intelligence. pp. 492–495. Springer (2012)

[37] Williamson, T.: Modal Logic as Metaphysics. Oxford University Press (2013)

[38] Wisniewski, M., Steen, A.: Embedding of Quantified Higher-Order Nominal Modal Logic into Classical Higher-Order Logic. In: Benzmüller, C., Otten, J. (eds.) Automated Reasoning in Quantified Non-Classical Logics (ARQNL), Proceedings. EPiC, vol. 33, pp. 59–64. EasyChair (2014)

[39] Wisniewski, M., Steen, A., Benzmüller, C.: TPTP and beyond: Representation of quantified non-classical logics. In: Benzmüller, C., Otten, J. (eds.) ARQNL 2016. Automated Reasoning in Quantified Non-Classical Logics. vol. 1770, pp. 51–65. CEUR Workshop Proceedings (2016)