Embedding and Automating Conditional Logics in Classical Higher-Order Logic

Christoph Benzmüller[1], Dov Gabbay[2],
Valerio Genovese[3,4], and Daniele Rispoli[4]

[1] Freie Universität Berlin, Germany
[2] King's College London, UK
[3] University of Luxembourg, Luxembourg
[4] University of Torino, Italy

Abstract. A sound and complete embedding of conditional logics into classical higher-order logic is presented. This embedding enables the application of off-the-shelf higher-order automated theorem provers and model finders for reasoning within and about conditional logics.

1 Introduction

Conditional logics capture default entailment in a modal framework via a defeasible implication operator "$\Rightarrow$" such that $\alpha \Rightarrow \beta$ reads as, "If α then, typically β". A peculiarity of conditional logics is that α is a formula and can contain other occurrences of "$\Rightarrow$".

Thanks to their expressivity, conditional logics have been successfully applied in several domains like non-monotonic reasoning [9], belief revision [12] and security [13].

Despite their wide range of potential applications, the formalization of a proper proof theory for conditional logics has been tackled only recently and only for a limited set of axiomatizations [18,19]. Moreover, there is no uniform framework to specify and reason about such formalisms.

Following the work of [6], a semantic embedding of conditional logic in classical higher-order logic HOL (Church's type theory) is presented. This embedding exploits the natural correspondence between selection function semantics for conditional logics [20] and HOL. In fact, selection function semantics can be seen as an higher-order extension of well-known Kripke semantics for modal logic and cannot be naturally embedded into first-order logic.

The contributions of the paper are threefold. First, we prove that the presented embedding is *sound* and *complete* w.r.t. selection function semantics. Second, we show how to apply off-the-shelf higher-order theorem provers and model finders for reasoning *within* and *about* conditional logics. Third, we investigate the practical value of such embedding through several experiments with different higher-order reasoning systems (HOL-RSs). As part of these experiments, several correspondence results between prominent conditional logic axioms and related semantic conditions have been automatically verified.

2 Conditional Logics

In order to make the paper self contained, we briefly resume syntax and semantics of conditional logics. For a deeper treatment we refer to [18].

Definition 1. *The* formulas *of conditional logic are given by*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \Rightarrow \varphi$$

where p ranges over a set of Boolean variables and $\Rightarrow$ is a binary modal operator.

From the selected set of primitive connectives, other logical connectives can be introduced as abbreviations: e.g., $\varphi \wedge \psi$ and $\varphi \rightarrow \psi$ (material implication) abbreviate $\neg(\neg\varphi \vee \neg\psi)$ and $\neg\varphi \vee \psi$, etc. Syntactically, conditional logics can be seen as a generalization of multimodal logic where the index of modality $\Rightarrow$ is a formula of the same language. For instance, in $(A \Rightarrow B) \Rightarrow C$ the subformula $A \Rightarrow B$ is the index of the second occurrence of $\Rightarrow$.

Regarding semantics, many different formalizations have been proposed (see [15]), here we focus on the *selection function semantics* [10], which is based on possible world structures and has been successfully used in [17] to develop proof methods for some conditional logics.

Definition 2. *A* model *of conditional logics is a tuple $\mathcal{M} = \langle S, f, h \rangle$ where,*

- *S is a non empty set of items called states;*
- *$f : S \times 2^S \rightarrow 2^S$ is the selection function;*
- *h is an assignment which, for each Boolean variable p, assigns the subset of states $h(p)$ where p holds.*

Intuitively the selection function f selects, for a world w and a formula φ, the set of worlds $f(w, A)$ which are "most-similar to w" or "closer to w" given the information φ.

Definition 3 (Semantic Interpretation). *An interpretation for a conditional logic is a pair $\mathcal{M}, s$ where $\mathcal{M}$ is a model and s is a state in $\mathcal{M}$. The satisfaction relation $\models$ holds between interpretations and formulae of the logic, and it is defined recursively as follows:*

- *$\mathcal{M}, s \models p$ iff $s \in h(p)$*
- *$\mathcal{M}, s \models \neg\varphi$ iff not $\mathcal{M}, s \models \varphi$*
- *$\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$*
- *$\mathcal{M}, s \models \varphi \Rightarrow \psi$ iff $\mathcal{M}, t \models \psi$ for all $t \in f(s, [\varphi])$ where, $[\varphi] = \{w \mid \mathcal{M}, w \models \varphi\}$.*

As usual, a conditional formula φ is *valid in a model* $\mathcal{M} = \langle S, f, h \rangle$, denoted with $\mathcal{M} \models \varphi$, iff for all $s \in S$ holds $\mathcal{M}, s \models \varphi$. A formula φ is *valid*, denoted $\models \varphi$, iff it is valid in every model.

Notice that f is defined to take $[\varphi]$ (called the *proof set* of φ w.r.t. a given model $\mathcal{M}$) instead of φ. This approach has the consequence of forcing the so-called *normality* property: given a model $\mathcal{M}$, if φ and φ' are equivalent (i.e.,

they are satisfied in the same set of states), then they index the same formulas w.r.t. to the $\Rightarrow$ modality.

The axiomatic counterpart of the normality condition is given by the rule (RCEA)

$$\frac{\varphi \leftrightarrow \varphi'}{(\varphi \Rightarrow \psi) \leftrightarrow (\varphi' \Rightarrow \psi)} \, (RCEA)$$

Moreover, it can be easily shown that the above semantics forces also the following rules to hold:

$$\frac{(\varphi_1 \wedge \ldots \wedge \varphi_n) \leftrightarrow \psi}{(\varphi_0 \Rightarrow \varphi_1 \wedge \ldots \wedge \varphi_0 \Rightarrow \varphi_n) \rightarrow (\varphi_0 \Rightarrow \psi)} \, (RCK)$$

$$\frac{\varphi \leftrightarrow \varphi'}{(\psi \Rightarrow \varphi) \leftrightarrow (\psi \Rightarrow \varphi')} \, (RCEC)$$

We refer to CK [10] as the minimal conditional logic closed under rules RCEA, RCEC and RCK. In what follows, only conditional logics extending CK are considered.

3 Classical Higher-Order Logic

HOL is a logic based on simply typed λ-calculus [11,2]. The set $\mathcal{T}$ of simple types in HOL is usually freely generated from a set of basic types $\{o, i\}$ (where o denotes the type of Booleans) using the function type constructor $\rightarrow$.

Definition 4. *The* terms *of HOL are defined by* $(\alpha, \beta, o \in \mathcal{T})$

$$s, t ::= p_\alpha \mid X_\alpha \mid (\lambda X_\alpha . s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta})_\beta \mid (\neg_{o \rightarrow o} s_o)_o \mid$$
$$(s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid (\Pi_{(\alpha \rightarrow o) \rightarrow o} s_{\alpha \rightarrow o})_o$$

p_α *denotes typed constants and* X_α *typed variables (distinct from* p_α*).*

Complex typed terms are constructed via abstraction and application. The primitive logical connectives are $\neg_{o \rightarrow o}$, $\vee_{o \rightarrow o \rightarrow o}$ and $\Pi_{(\alpha \rightarrow o) \rightarrow o}$ (for each type α). From these, other logical connectives can be introduced as abbreviations: e.g., $\wedge$ and $\rightarrow$ abbreviate the terms $\lambda A.\lambda B.\neg(\neg A \vee \neg B)$ and $\lambda A.\lambda B.\neg A \vee B$, etc. HOL terms of type o are called formulas. Binder notation $\forall X_\alpha . s_o$ is used as an abbreviation for $(\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda X_\alpha . s_o))$. Substitution of a term A_α for a variable X_α in a term B_β is denoted by $[A/X]B$, where it is assumed that the bound variables of B avoid variable capture. Well known operations and relations on HOL terms include $\beta\eta$-normalization and $\beta\eta$-equality, denoted by $s =_{\beta\eta} t$.

The following definition of HOL semantics closely follows the standard literature [1,2].

Definition 5. *A* frame *is a collection* $\{D_\alpha\}_{\alpha \in \mathcal{T}}$ *of nonempty sets called* do-mains *such that* $D_o = \{T, F\}$ *where* T *represents truth and* F *falsehood,* $D_i \neq \emptyset$ *is chosen arbitrary, and* $D_{\alpha \to \beta}$ *are collections of functions mapping* D_α *into* D_β.

Definition 6. *An* interpretation *is a tuple* $\langle \{D_\alpha\}_{\alpha \in \mathcal{T}}, I \rangle$ *where* $\{D_\alpha\}_{\alpha \in \mathcal{T}}$ *is a frame and where function* I *maps each typed constant* c_α *to an appropriate element of* D_α, *which is called the* denotation of c_α. *The denotations of* $\neg, \vee$ *and* $\Pi_{(\alpha \to o) \to o}$ *are always chosen as usual. A variable assignment* ϕ *maps variables* X_α *to elements in* D_α. *An interpretation is a* Henkin model (general model) *if and only if there is a binary valuation function* $\mathcal{V}$ *such that* $\mathcal{V}(\phi, s_\alpha) \in D_\alpha$ *for each variable assignment* ϕ *and term* s_α, *and the following conditions are satisfied for all* ϕ, *variables* X_α, *constants* p_α, *and terms* $l_{\alpha \to \beta}, r_\alpha, s_\beta$ *(for* $\alpha, \beta \in \mathcal{T}$*):*

- $\mathcal{V}(\phi, X_\alpha) = \phi(X_\alpha)$
- $\mathcal{V}(\phi, p_\alpha) = I(p_\alpha)$
- $\mathcal{V}(\phi, (l_{\alpha \to \beta}\ r_\alpha)) = (\mathcal{V}(\phi, l_{\alpha \to \beta}))(\mathcal{V}(\phi, r_\alpha))$
- $\mathcal{V}(\phi, \lambda X_\alpha.s_\beta)$ *represents the function from* D_α *into* D_β *whose value for each argument* $z \in D_\alpha$ *is* $\mathcal{V}(\phi[z/X_\alpha], s_\beta)$, *where* $\phi[z/X_\alpha]$ *is that variable assignment such that* $\phi[z/X_\alpha](X_\alpha) = z$ *and* $\phi[z/X_\alpha]Y_\beta = \phi Y_\beta$ *when* $Y_\beta \neq X_\alpha$.

If an interpretation $\mathcal{H} = \langle \{D_\alpha\}_{\alpha \in \mathcal{T}}, I \rangle$ is an *Henkin model* the function $\mathcal{V}$ is uniquely determined and $\mathcal{V}(\phi, s_\alpha) \in D_\alpha$ is called the denotation of s_α. $\mathcal{H}$ is called a *standard model* if and only if for all α and β, $D_{\alpha \to \beta}$ is the set of all functions from D_α into D_β. It is easy to verify that each standard model is also a Henkin model. A formula A of HOL is *valid* in a Henkin model $\mathcal{H}$ if and only if $\mathcal{V}(\phi, A) = T$ for all variable assignments ϕ. In this case we write $\mathcal{H} \models A$. A is (Henkin) valid, denoted as $\models A$, if and only if $\mathcal{H} \models A$ for all Henkin models $\mathcal{H}$.

Proposition 1. *Let* $\mathcal{V}$ *be the valuation function of Henkin model* $\mathcal{H}$. *The following properties hold for all assignments* ϕ, *terms* $s_o, t_o, l_\alpha, r_\alpha$, *and variables* X_α, V_α *(for* $\alpha \in \mathcal{T}$*):*

- $\mathcal{V}(\phi, (\neg s_o)) = T$ *iff* $\mathcal{V}(\phi, s_o) = F$
- $\mathcal{V}(\phi, (s_o \vee t_o)) = T$ *iff* $\mathcal{V}(\phi, s_o) = T$ *or* $\mathcal{V}(\phi, s_o) = T$
- $\mathcal{V}(\phi, (s_o \wedge t_o)) = T$ *iff* $\mathcal{V}(\phi, s_o) = T$ *and* $\mathcal{V}(\phi, s_o) = T$
- $\mathcal{V}(\phi, (s_o \to t_o)) = T$ *iff* $\mathcal{V}(\phi, s_o) = F$ *or* $\mathcal{V}(\phi, s_o) = T$
- $\mathcal{V}(\phi, (\forall X_\alpha.s_o)) = \mathcal{V}(\phi, (\Pi_{(\alpha \to o) \to o}\ (\lambda X_\alpha.s_o))) = T$ *iff for all* $v \in D_\alpha$ *holds* $\mathcal{V}(\phi[v/V_\alpha], ((\lambda X_\alpha.s_o)\ V)) = T$
- *if* $l_\alpha =_{\beta\eta} r_\alpha$ *then* $\mathcal{V}(\phi, l_\alpha) = \mathcal{V}(\phi, r_\alpha)$

4   Embedding Conditional Logics in HOL

Conditional logic formulas are identified with certain HOL terms (predicates) of type $i \to o$. They can be applied to terms of type i, which are assumed to denote possible states.

Definition 7. *The mapping $\lfloor \cdot \rfloor$ translates formulas φ of conditional logic CK into HOL terms $\lfloor \varphi \rfloor$ of type $i \to o$. The primitives of conditional logic are mapped as follows:*

$$
\begin{aligned}
\lfloor p \rfloor &= p_{i \to o} && (p_{i \to o} \text{ is a HOL constant symbol}) \\
\lfloor \neg \rfloor &= \lambda A_{i \to o}.\lambda X_i.\neg(A\ X) \\
\lfloor \vee \rfloor &= \lambda A_{i \to o}.\lambda B_{i \to o}.\lambda X_i.(A\ X) \vee (B\ X) \\
\lfloor \Rightarrow \rfloor &= \lambda A_{i \to o}.\lambda B_{i \to o}.\lambda X_i.\forall W_i.(f\ X\ A\ W) \to (B\ W)
\end{aligned}
$$

The constant symbol f in the mapping of $\Rightarrow$ is of type $i \to (i \to o) \to (i \to o)$. It realizes the selection function, i.e., its interpretation is chosen appropriately (cf. below).

Compound formulas are recursively mapped as follows:

$$
\begin{aligned}
\lfloor \neg\varphi \rfloor &= (\lfloor \neg \rfloor \lfloor \varphi \rfloor) \\
\lfloor \varphi \vee \psi \rfloor &= (\lfloor \vee \rfloor \lfloor \varphi \rfloor \lfloor \psi \rfloor) \\
\lfloor \varphi \wedge \psi \rfloor &= (\lfloor \wedge \rfloor \lfloor \varphi \rfloor \lfloor \psi \rfloor) \\
\lfloor \varphi \Rightarrow \psi \rfloor &= (\lfloor \Rightarrow \rfloor \lfloor \varphi \rfloor \lfloor \psi \rfloor)
\end{aligned}
$$

Analyzing the validity of a translated formula $\lfloor \varphi \rfloor$ for a state represented by term t_i corresponds to evaluating the application $(\lfloor \varphi \rfloor\ t_i)$. In line with [6], we can easily encode the notion of validity as follows

$$
vld := \lambda A_{i \to o}.\forall S_i.(A\ S)
$$

With this definition, validity of a conditional formula φ in CK corresponds to the validity of the corresponding formula $(vld\ \lfloor \varphi \rfloor)$ in HOL, and vice versa.

We illustrate the approach with formula $p \Rightarrow p$ where p is a Boolean variable. This formula corresponds to the HOL term $(vld\ \lfloor p \Rightarrow p \rfloor)$ which expands into (type information is omitted) $(\lambda A.\forall S.(A\ S))((\lambda A.\lambda B.\lambda X.\forall W.(f\ X\ A\ W) \to (B\ W))\ p\ p)$ and $\beta\eta$-normalizes to $\forall S.\forall W.(f\ S\ p\ W) \to (p\ W))$. It is easy to verify that this HOL formula is countersatisfiable, which is the expected result in CK.

To prove the soundness and completeness of the embedding, a mapping from selection function models into Henkin models is employed.

Definition 8. *Given a selection function model $\mathcal{M} = \langle S, f, \models \rangle$. The Henkin model $\mathcal{H}^{\mathcal{M}} = \langle \{D_\alpha\}_{\alpha \in \mathcal{T}}, I \rangle$ for $\mathcal{M}$ is defined as follows: D_i is chosen as the set of states S, and for all combinations of α and β, $D_{\alpha \to \beta}$ is chosen as the set of all functions from D_α to D_β.[1] Let $p^1, \ldots, p^m$ for $m \geq 1$ be the Boolean variables of the conditional logic and let $\lfloor p^j \rfloor = p^j_{i \to o}$ for $i = 1, \ldots, m$. We define I as follows:*

---

[1] This choice in particular means that $D_{i \to o}$ is the set of all possible predicates q over S; these predicates can also be viewed a sets $\{x \in S \mid q(x) = T\}$. Note, that modulo this technicality, $D_{i \to o}$ is identical to '2^S' in Def. 2.

- For $1 \leq i \leq m$, choose $I(p_{i\to o}^j) \in D_{i\to o}$ so that $(I(p_{i\to o}^j))(w) = T$ for all $w \in D_i$ with $\mathcal{M}, w \models p^j$, and $(I(p_{i\to o}^j))(w) = F$ otherwise.
- Choose $I(f_{i\to(i\to o)\to(i\to o)}) \in D_{i\to(i\to o)\to(i\to o)}$ so that for all $s, t \in D_i$ and $q \in D_{i\to o}$ holds $(I(f_{i\to(i\to o)\to(i\to o)}))(s, q, t) = T$ if $t \in f(s, \{x \in S \mid q(x) = T\})$ in $\mathcal{M}$, and $(I(f_{i\to(i\to o)\to(i\to o)}))(s, q, t) = F$ otherwise.
- For all other constants s_α, choose $I(s_\alpha)$ arbitrarily.[2]

It is easy to verify that $\mathcal{H}^{\mathcal{M}}$ is a Henkin model.[3] It is even a standard model, since the function spaces are full.

Lemma 1. *Let $\mathcal{H}^{\mathcal{M}}$ be a Henkin model for a selection function model $\mathcal{M}$. For all conditional logic formulas φ, states s, and variable assignments ϕ it holds:*

$$\mathcal{M}, s \models \varphi \quad \textit{iff} \quad \mathcal{V}(\phi[s/S_i], (\lfloor \varphi \rfloor \; S)) = T$$

Proof. The proof is by induction on the structure of φ.

$\varphi = p^j$. By definition of $\lfloor \cdot \rfloor$, $\mathcal{V}$, and $\mathcal{H}^{\mathcal{M}}$ holds $\mathcal{V}(\phi[s/S_i], (\lfloor p^j \rfloor \; S)) = \mathcal{V}(\phi[s/S_i], (p_{i\to o}^j \; S)) = (I(p_{i\to o}^j))(s) = T$ iff $\mathcal{M}, s \models p^j$.

For $\varphi = (\neg r)$ the argument is similar to $\varphi = (p \vee r)$ below.

$\varphi = (p \vee r)$. $\mathcal{M}, s \models (p \vee t)$ iff $\mathcal{M}, s \models p$ or $\mathcal{M}, s \models t$. By induction $\mathcal{V}(\phi[s/S_i], (\lfloor p \rfloor \; S)) = T$ or $\mathcal{V}(\phi[s/S_i], (\lfloor r \rfloor \; S)) = T$. By Prop. 1, definition of $\lfloor \cdot \rfloor$ and since $((\lfloor p \vee r \rfloor) \; S) =_{\beta\eta} ((\lfloor p \rfloor \; S) \vee (\lfloor r \rfloor \; S))$ it holds $\mathcal{V}(\phi[s/S_i], ((\lfloor p \vee r \rfloor) \; S)) = \mathcal{V}(\phi[s/S_i], ((\lfloor p \rfloor \; S) \vee (\lfloor r \rfloor \; S))) = T$.

$\varphi = (p \Rightarrow r)$. $\mathcal{M}, s \models p \Rightarrow r$ iff, for all $t \in f(s, [p])$ holds $\mathcal{M}, t \models r$. This is equivalent to, for all t holds (i) $t \notin f(s, [p])$ or (ii) $\mathcal{M}, t \models r$. By induction applied to (ii) with $\phi' = \phi[s/S_i]$ holds $\mathcal{V}(\phi'[t/T_i], (\lfloor r \rfloor \; T)) = T$. Furthermore, again by induction, for all t and ϕ'' with $\phi'' = \phi'[t/T_i] = \phi[s/S_i][t/T_i]$ we have $\mathcal{M}, u \models p$ iff $\mathcal{V}(\phi''[u/U_i], (\lfloor p \rfloor \; U_i)) = T$. Hence, by construction of $\mathcal{H}^{\mathcal{M}}$, (i) is equivalent to $(I(f))(s, [p], t) = (I(f))(s, \{u \mid \mathcal{M}, u \models p\}, t) = (I(f))(s, \{u \mid \mathcal{V}(\phi''[u/U_i], (\lfloor p \rfloor \; U_i)) = T\}, t) = F$. By definition of $\mathcal{V}$, and since $s = \mathcal{V}(\phi'', S_i)$ and $t = \mathcal{V}(\phi'', T_i)$ it holds $\mathcal{V}(\phi'', (f \; S_i \; \lfloor p \rfloor \; T_i)) = F$. By combining these results and by Prop. 1 we get that for all t $\mathcal{V}(\phi'[t/T_i], ((f \; S_i \; \lfloor p \rfloor \; T_i) \to (\lfloor r \rfloor \; T_i))) = \mathcal{V}(\phi'[t/T_i], (\lambda W.((f \; S_i \; \lfloor p \rfloor \; W_i) \to (\lfloor r \rfloor \; W_i)) \; T_i)) = T$, which by Prop. 1 is equivalent to $\mathcal{V}(\phi', (\Pi_{(i\to o)\to o} \; (\lambda W_i.((f \; S_i \; \lfloor p \rfloor \; W_i) \to (\lfloor r \rfloor \; W_i))))) = \mathcal{V}(\phi', (\forall W_i.((f \; S_i \; \lfloor p \rfloor \; W_i) \to (\lfloor r \rfloor \; W_i))) = T$. By Prop. 1, definition of $\lfloor \cdot \rfloor$ and since $(\lfloor p \Rightarrow r \rfloor \; S) =_{\beta\eta} (\forall W_i.((f \; S_i \; \lfloor p \rfloor \; W_i) \to (\lfloor r \rfloor \; W_i)))$ we finally have $\mathcal{V}(\phi[s/S_i], (\lfloor p \Rightarrow r \rfloor \; S)) = T$.

---

[2] In fact, we may safely assume that there are no other typed constant symbols given, except for the symbol $f_{i\to(i\to o)\to(i\to o)}$, the symbols $p_{i\to o}^j$, and the logical connectives.

[3] In $\mathcal{H}^{\mathcal{M}}$ we have merely fixed D_i and the interpretation of the constant symbols $p_{i\to o}^j$ and $f_{i\to(i\to o)\to(i\to o)}$. These constraints are obviously not in conflict with any of the requirements in Defs. 5 and 6. The existence of a valuation function $\mathcal{V}$ for an HOL interpretation crucially depends on how sparse the function spaces have been chosen in frame $\{D_\alpha\}_{\alpha \in \mathcal{T}}$. [1] discusses criteria that are sufficient to ensure the existence of a valuation function; they require that certain λ-abstractions have denotations in frame $\{D_\alpha\}_{\alpha \in \mathcal{T}}$. The function spaces are full, so this is trivially the case.

Theorem 1 (Soundness and Completeness).

$$\models (vld \lfloor \varphi \rfloor) \ in \ HOL \quad if \ and \ only \ if \quad \models \varphi \ in \ CK$$

Proof. (Soundness) The proof is by contraposition. Assume $\not\models \varphi$ in CK, that is, there is a selection function model $\mathcal{M} = \langle S, f, h \rangle$ and a state $s \in S$, such that $\mathcal{M}, s \not\models \varphi$. By Lemma 1 we have that $\mathcal{V}(\phi[s/S_i], (\lfloor \varphi \rfloor \ S)) = F$ for a variable assignment ϕ in Henkin model $\mathcal{H}^{\mathcal{M}} = \langle \{D_\alpha\}_{\alpha \in \mathcal{T}}, I \rangle$ for $\mathcal{M}$. Thus, by Prop. 1, definition of vld and since $(\forall S_i . \lfloor \varphi \rfloor \ S) =_{\beta\eta} (vld \lfloor \varphi \rfloor)$ we know that $\mathcal{V}(\phi, (\forall S_i . \lfloor \varphi \rfloor \ S)) = \mathcal{V}(\phi, (vld \lfloor \varphi \rfloor)) = F$. Hence, $\mathcal{H}^{\mathcal{M}} \not\models (vld \lfloor \varphi \rfloor)$, and thus $\not\models (vld \lfloor \varphi \rfloor)$ in HOL.

(Completeness) The proof is again by contraposition. Assume $\not\models (vld \lfloor \varphi \rfloor)$ in HOL, that is, there is a Henkin model $\mathcal{H} = \langle \{D_\alpha\}_{\alpha \in \mathcal{T}}, I \rangle$ and a variable assignment ϕ with $\mathcal{V}(\phi, (vld \lfloor \varphi \rfloor)) = F$. From $(vld \lfloor \varphi \rfloor) =_{\beta\eta} (\forall S_i . \lfloor \varphi \rfloor \ S)$ and Prop. 1 we get $\mathcal{V}(\phi, (\forall S_i . \lfloor \varphi \rfloor \ S)) = F$, and hence, by definition of vld, $\mathcal{V}(\phi[s/S_i], \lfloor \varphi \rfloor \ S) = F$ for some $s \in D_i$. Without loss of generality we can assume that Henkin Model $\mathcal{H}$ is in fact a Henkin model $\mathcal{H}^{\mathcal{M}}$ for a corresponding selection function model $\mathcal{M}$. By Lemma 1 we thus know that $\mathcal{M}, s \not\models \varphi$, and hence $\not\models \varphi$ in CK.

Theorem 1 does not trivially follow from previous literature on embedding modal logics into HOL because of the complexity of the selection function. In fact, standard modalities are usually evaluated over a so-called accessibility relationship of type $R(i, w)$, where i is an index and w is a world. Conditional modalities are instead evaluated over selection functions of type $f(w, A)$ where w is a world, A is a *set of worlds* and f is a *function* which returns a set of worlds.

5   Experiments: Analyzing the Literature

The presented semantic embedding of conditional logics into HOL is of practical relevance. It supports the application of standard HOL-RSs to problems encoded *within* conditional logics and also to problems *about* conditional logics. Examples of the latter kind include correspondence claims between certain axioms and related conditions on the semantic structures (e.g., the conditional logic axiom $\varphi \Rightarrow \varphi$ corresponds to the semantic condition that $f(w, [\varphi]) \subseteq [\varphi]$).

This section reports on experiments in which such kind of questions have been studied with HOL-RSs. The HOL-RSs employed in the case study are:

LEO-II (version v1.2.6). A higher-order automated theorem prover based on extensional resolution [7]. LEO-II[4] cooperates with the first-order theorem prover E.

TPS (version 3.080227G1d). A fully automated version of the higher-order theorem proving environment TPS[5] [3]. Proof search in TPS is controlled by modes (sets of flag settings), and the automated TPS version employed here applies strategy scheduling over these modes.

---

[4] `http://www.leoprover.org`

[5] `http://gtps.math.cmu.edu/tps.html`

Satallax (version 1.4). A higher-order automated theorem prover based on a complete ground tableau calculus for HOL with a choice operator [4]. Satallax[6], which cooperates with SAT solver MiniSat, has additional model finding capabilities.

IsabelleP (version 2009-2). The proof assistant Isabelle/HOL[7] [14] is normally used interactively. IsabelleP is an automated version of Isabelle/HOL, in which several tactics are subsequently applied.

Refute and Nitpick (versions 2009-2). Isabelle/HOL's ability to find (counter-)models using the *refute* [24] and *nitpick*[8] [8] commands has been integrated into automatic systems.

The reasoning systems described above are available online via the System-OnTPTP tool [21]. They support the new TPTP THF infrastructure for HOL [22] and they accept problems formalized in the THF representation language.

The problems studied in the experiments are:

Problem 1. Is the presented embedding consistent? In order to study this question, the formalization of the embedding has been passed to the HOL-RSs.

Problem 2. Are the rules RCEA, RCK, and RCEC implied in the embedding? They obviously should, since CK is defined as the minimal conditional logic closed under these rules. The problems passed to the HOL-RSs are (types are omitted): RCEA: $\forall P, Q, R.(\text{vld } \lfloor P \leftrightarrow Q \rfloor) \to (\text{vld } \lfloor (P \Rightarrow R) \leftrightarrow (Q \Rightarrow R) \rfloor)$, RCK: $\forall P0, P1, P2, Q.(\text{vld } \lfloor (P1 \land P2) \leftrightarrow Q \rfloor) \to (\text{vld } \lfloor ((P0 \Rightarrow P1) \land (P0 \Rightarrow P2)) \to (P0 \Rightarrow Q) \rfloor)$, RCEC: $\forall P, Q, R.(\text{vld } \lfloor P \leftrightarrow Q \rfloor) \to (\text{vld } \lfloor (R \Rightarrow P) \leftrightarrow (R \Rightarrow Q) \rfloor)$

Problem 3. Do the correspondence results between conditional logic axioms and semantic conditions as presented in Figure 1 (copied from [18]) indeed hold?

ID $(\forall A.\text{vld } \lfloor A \Rightarrow A \rfloor) \leftrightarrow (\forall A, W.(f \ W \ A) \subseteq A)$
MP $(\forall A, B.\text{vld } \lfloor (A \Rightarrow B) \to (A \to B) \rfloor) \leftrightarrow (\forall A, W.(A \ W) \to ((f \ W \ A) \ W))$
CS $(\forall A, B.\text{vld } \lfloor (A \land B) \to (A \Rightarrow B) \rfloor) \leftrightarrow (\forall A, W.(A \ W) \to (f \ W \ A) \subseteq (\lambda X_i.X = W))$
CEM $(\forall A, B.\text{vld } \lfloor (A \Rightarrow B) \lor (A \Rightarrow \neg B) \rfloor) \leftrightarrow (\forall A, W.(f \ W \ A) = \emptyset \lor \exists V.(f \ W \ A) = (\lambda X.X = V))$
... The formalizations of AC, RT, CV, and CA are analogous, and the HOL encodings of $\subseteq, \emptyset, \cap,$ and $\cup$ are straightforward.

In the experiments, each equivalence statement has actually been split in its two implication directions.

Problem 4. A subtle point, concerning correspondence theory for conditional logics, is the interpretation of the scopes of the implicit universal quantifiers

---

[6] http://www.ps.uni-saarland.de/~cebrown/satallax/
[7] http://isabelle.in.tum.de
[8] http://www4.in.tum.de/~blanchet/nitpick.html

ID	Axiom	$A \Rightarrow A$
	Condition	$f(w, [A]) \subseteq [A]$
MP	Axiom	$(A \Rightarrow B) \rightarrow (A \rightarrow B)$
	Condition	$w \in [A] \rightarrow w \in f(w, [A])$
CS	Axiom	$(A \wedge B) \rightarrow (A \Rightarrow B)$
	Condition	$w \in [A] \rightarrow f(w, [A]) \subseteq \{w\}$
CEM	Axiom	$(A \Rightarrow B) \vee (A \Rightarrow \neg B)$
	Condition	$|f(w, [A])| \leq 1$
AC	Axiom	$(A \Rightarrow B) \wedge (A \Rightarrow C) \rightarrow (A \wedge C \Rightarrow B)$
	Condition	$f(w, [A]) \subseteq [B] \rightarrow f(w, [A \wedge B]) \subseteq f(w, [A])$
RT	Axiom	$(A \wedge B \Rightarrow C) \rightarrow ((A \Rightarrow B) \rightarrow (A \Rightarrow C))$
	Condition	$f(w, [A]) \subseteq [B] \rightarrow f(w, [A]) \subseteq f(w, [A \wedge B])$
CV	Axiom	$(A \Rightarrow B) \wedge \neg(A \Rightarrow \neg C) \rightarrow (A \wedge C \Rightarrow B)$
	Condition	$(f(w, [A]) \subseteq [B] \text{ and } f(w, [A]) \cap [C] \neq \emptyset) \rightarrow f(w, [A \wedge C]) \subseteq [B]$
CA	Axiom	$(A \Rightarrow B) \wedge (C \Rightarrow B) \rightarrow (A \vee C \Rightarrow B)$
	Condition	$f(w, [A \vee B]) \subseteq f(w, [A]) \cup f(w, [B])$

Fig. 1. Conditional logic axioms and semantic conditions

in the correspondence statements in Figure 1. For example, for ID and MP we might read (ID') $\forall A.((\text{vld } \lfloor A \Rightarrow A \rfloor) \leftrightarrow W.((f\ W\ A) \subseteq A))$ and (MP') $\forall A, B.(\text{vld } \lfloor (A \Rightarrow B) \rightarrow (A \rightarrow B) \rfloor \leftrightarrow (\forall W.(A\ W) \rightarrow ((f\ W\ A)\ W)))$. An interesting, non-trivial question (suited also for sharpening the intuition on the particular conditional logics axioms) is whether these misread statements are still provable. Therefore analogous primed versions have been formalized for all correspondence problems as further benchmark examples.

Problem 5. Do the following logic inclusions hold: (a) CK+{MP,CS} includes CK+{CEM}? (b) CK+{CEM,MP} includes CK+{CS}? (c) CK+{RT,AC} includes CK+$\{(A \Rightarrow B) \rightarrow (((A \wedge B) \Rightarrow C) \leftrightarrow (A \Rightarrow C))\}$? The formalizations are obvious and we show only the case for (a):

$$\forall A, B.\text{vld } \lfloor (A \Rightarrow B) \rightarrow (A \rightarrow B) \rfloor,$$
$$\forall A, B.\text{vld } \lfloor (A \wedge B) \rightarrow (A \Rightarrow B) \rfloor$$
$$\vdash \quad \forall A, B.\text{vld } \lfloor (A \Rightarrow B) \vee (A \Rightarrow \neg B) \rfloor$$

With the results from Problem (3), such questions can alternatively be formalized with the respective semantic conditions.

The detailed results of the experiments are presented in Table 1. Exploiting the SystemOnTPTP infrastructure, all experiment runs were done remotely at the University of Miami on 2.80GHz computers with 1GB memory and running the Linux operating system. The timeout was set to 180 seconds.

The first column of the table presents the problem number and the second column presents the result status as confirmed by the HOL-RSs: THM stands for 'theorem', CSA for 'countersatisfiable', and SAT for 'satisfiable'. The remaining

Problem	Status	L	T	S	I	N	R
1	SAT			0.26		3.34	2.88
2_RCEA	THM	0.06	0.44	0.29	18.02		
2_RCK	THM	0.06	0.36	0.29	18.03		
2_RCEC	THM	0.05	0.36	0.30	32.94		
3_ID$^{\leftarrow}$	THM	0.03	0.36	0.27	18.03		
3_ID$^{\rightarrow}$	THM	0.03	0.34	0.27	17.95		
3_MP$^{\leftarrow}$	THM	0.04	0.34	0.27	18.18		
3_MP$^{\rightarrow}$	THM	0.04	0.35		18.05		
3_CS$^{\leftarrow}$	THM	0.04	0.37	0.27	48.13		
3_CS$^{\rightarrow}$	THM	0.13	0.37	0.27	48.42		
3_CEM$^{\leftarrow}$	THM		1.04		33.24		
3_CEM$^{\rightarrow}$	THM	0.13	0.38		34.10		
3_AC$^{\leftarrow}$	THM		0.38	0.30	21.85		
3_AC$^{\rightarrow}$	THM		0.40		18.05		
3_RT$^{\leftarrow}$	THM		0.62	1.33	18.05		
3_RT$^{\rightarrow}$	THM		0.42		18.08		
3_CV$^{\leftarrow}$	THM		0.44		125.35		
3_CV$^{\rightarrow}$	THM		0.44	86.83			
3_CA$^{\leftarrow}$	THM		0.37	1.44	18.06		
3_CA$^{\rightarrow}$	THM		6.83				
4_ID'$^{\leftarrow}$	THM	0.05	0.50	0.37	18.05		
4_ID'$^{\rightarrow}$	THM	0.08	0.44	0.36	18.20		
4_MP'$^{\leftarrow}$	THM	0.06	0.49	0.34	17.98		
4_MP'$^{\rightarrow}$	CSA					3.45	3.00
4_CS'$^{\leftarrow}$	THM	0.04	0.36	0.29	18.01		
4_CS'$^{\rightarrow}$	CSA			0.28		3.56	3.11
4_CEM'$^{\leftarrow}$	THM		1.04	2.00	33.23		
4_CEM'$^{\rightarrow}$	CSA					3.59	2.98
4_AC'$^{\leftarrow}$	?						
4_AC'$^{\rightarrow}$	CSA					4.75	3.82
4_RT'$^{\leftarrow}$	THM	109.42	31.73	0.39	18.05		
4_RT'$^{\rightarrow}$	CSA					3.60	3.00
4_CV'$^{\leftarrow}$	THM	111.06	31.85	0.40	51.14		
4_CV'$^{\rightarrow}$	THM		31.78	0.41	54.61		
4_CA'$^{\leftarrow}$	CSA					3.57	3.09
4_CA'$^{\rightarrow}$	?						
5(a)_ax.	CSA					3.69	3.01
5(a)_sem.	CSA					3.62	3.06
5(b)_ax.	THM	0.06	8.71				
5(b)_sem.	THM				63.30		
5(c)_ax.	THM		56.52				
5(c)_sem.	THM		14.89	44.69	42.46		

Table 1. Performance results of HOL-RSs.

columns report the time after which the particular HOL-RSs reported the displayed status. L stands for LEO-II, T for TPS, S for Satallax, I for IsabelleP, N for Nitpick, and R for Refute.

All correspondence claims have been confirmed by the HOL-RSs. For the primed versions the situation is different and several counterexamples have been reported by the model finders, in particular, for several forward directions. Two of these counterexamples are exemplary presented next. It is straightforward to check that they indeed invalidate the respective primed correspondence statements.[9]

Refute reports the following countermodel for $MP^{\rightarrow}$: choose $D_i = \{i1\}$, $A = \{i1\}$, $B = \{i1\}$, $W = i1$, and

$$f = \left\{ i1 \longrightarrow \begin{cases} \emptyset & \longrightarrow \emptyset \\ \{i1\} & \longrightarrow \emptyset \end{cases} \right.$$

Nitpick reports for $RT'^{\rightarrow}$: choose $D_i = \{i1, i2\}$, $A = \{i2\}$, $B = \{i1\}$, $C = \emptyset$, $W = i2$, and

$$f = \begin{cases} i1 \longrightarrow \begin{cases} \emptyset & \longrightarrow \emptyset \\ \{i1\} & \longrightarrow \emptyset \\ \{i2\} & \longrightarrow \{i2\} \\ \{i1, i2\} \longrightarrow \emptyset \end{cases} \\ i2 \longrightarrow \begin{cases} \emptyset & \longrightarrow \{i2\} \\ \{i1\} & \longrightarrow \emptyset \\ \{i2\} & \longrightarrow \{i1\} \\ \{i1, i2\} \longrightarrow \emptyset \end{cases} \end{cases}$$

For both the axiomatic and semantic formalization of Problem 5(a) countermodels are quickly found. For example, for the axiomatic version 5(a)_ax. Nitpick reports: choose $D_i = \{i1, i2\}$, $A = \emptyset$, $B = \{i1\}$, and

$$f = \begin{cases} i1 \longrightarrow \begin{cases} \emptyset & \longrightarrow \emptyset \\ \{i1\} & \longrightarrow \{i1\} \\ \{i2\} & \longrightarrow \emptyset \\ \{i1, i2\} \longrightarrow \emptyset \end{cases} \\ i2 \longrightarrow \begin{cases} \emptyset & \longrightarrow \{i1, i2\} \\ \{i1\} & \longrightarrow \emptyset \\ \{i2\} & \longrightarrow \{i2\} \\ \{i1, i2\} \longrightarrow \{i2\} \end{cases} \end{cases}$$

Inclusion claims 5(b) and 5(c) are confirmed as theorems.

With the tools provided by the SystemOnTPTP infrastructure it is straightforward to write a small shell script which bundles the mentioned HOL-RSs into

---

[9] Concerning, problem 4 we might wonder why the suggested denotations for f below cannot be used as candidates for generating countermodels to the corresponding non-primed correspondence statements — this is clearly not the case: e.g., note that the f suggested for invalidating $MP'^{\rightarrow}$ (which returns $\emptyset$ for all arguments W and A) is in fact incompatible with (and thus excluded by) $MP^{\rightarrow}$s antecedent (to see this choose $A = \{i1\}$, $B = \emptyset$). Such a kind of further analysis is again effectively supported by the HOL-RSs.

a single online reasoning system, and, in fact, this is how the experiments presented in this section have been carried out. As the results demonstrate, this combined HOL reasoner is powerful for reasoning about conditional logics; in particular the combination of HOL theorem proving and HOL (counter-)model finding is intriguing. Hence, there is good evidence that the HOL-RSs could fruitfully support the analysis of similar questions in the exploration of further conditional logics. Note also, that in most cases there are at least two matching results by independent systems. Another interesting observation is that TPS was the strongest prover in the experiments followed by IsabelleP, Satallax, and LEO-II. Since this is exactly the opposite order of the outcome of the 2010 CASC[10] competition, these problems are obviously interesting new benchmarks for the TPTP library.

The approach is applicable also to reasoning within conditional logics For example, formula $((p \Rightarrow q) \leftrightarrow (p \rightarrow q)) \rightarrow (p \Rightarrow p)$ is obviously countersatisfiable, and all model finders quickly find respective countermodels. Satallax is fastest in 0.28 seconds. The countermodel reported by Nitpick is $D_i = \{i1\}, p = \emptyset, q = \{i1\}$, and

$$f = \left\{ i1 \longrightarrow \begin{cases} \emptyset & \longrightarrow \{i1\} \\ \{i1\} & \longrightarrow \emptyset \end{cases} \right.$$

Unfortunately, a library of specific benchmark problems for conditional logics is currently not available, and therefore the (direct) conditional logics provers CondLean, GoalDuck and leanCK have been evaluated in [16] only with respect to classical modal logic problems. In this evaluation the modal logic problems were encoded in conditional logics by defining $\Box\varphi$ as an abbreviation for $T \Rightarrow \varphi$. Evaluating our approach wrt. these artificially encoded classical modal logic problems does hardly make sense though, and the existing direct embedding of classical modal logic in HOL [6] should for good reasons be preferred for these test examples. First experiments with small hand-translated examples from this test suite were nevertheless successful.

Evidence against the preconception that our higher-order based approach to reasoning in conditional logics cannot be effective in practice comes from a recent case study on automated reasoning in first-order modal logics [23]. In this case study an higher-order based approach which is closely related to the one presented here (and which was realized with the provers Satallax and LEO-II), performed reasonably well behind the specialist provers MLeanTAB[11] and MLeanSeP[12]. In particular, the higher-order provers did better (in terms of proving problems) than the direct prover GQML[13] and a first-order solution based on MSPASS[14].

As a final remark on this section we underline that solutions to problems in Section 5 are already known in theory of conditional logics. This does not straightforwardly imply that HOL-Reasoners (HOL-RSs) can solve them. In gen-

---

[10] http://www.tptp.org/CASC/J5/

[11] http://www.leancop.de/mleantap/programs/mleantap11.pl

[12] http://www.leancop.de/mleansep/programs/mleansep11.pl

[13] http://cialdea.dia.uniroma3.it/GQML/

[14] http://www.cs.man.ac.uk/~schmidt/mspass/

eral, HOL is undecidable, the aim of what presented above is to show that the theoretical embedding of CLs into HOL has also practical benefits due that it is possible to use HOL-RSs to reason about and within CLs. This is possible by making HOL-RSs cooperate working on the same task. Concerning model generation, we show that countermodels can be constructed for ill-formulated/ill-conjectured correspondence claims.

6   Conclusion

A sound and complete embedding of conditional logics into classical higher-order logic has been presented. Similar to other non-classical logics [6], conditional logics can be seen as natural fragments of classical higher-order logic, and they can be studied and automated as such. Up to authors knowledge, the presented work is the first integrated approach to automated deduction for *all* extensions of CK that involve *any* combination of the axioms reported in Figure 1. Previously existing proof methods [18,19] are limited to extensions of CK including only ID, MP, CS and CEM. Theorem proving for conditional logics appears to be much more difficult than for modal logic. There are very few modal provers for CLs, namely there are sequent calculi for CK+{ID,MP/CEM,CS} [18] and tableaux for CK+{CEM,MP} [19]. Model builders exist only for CK+{CEM,MP} [19]. No theorem provers are known for CK+{CS,AC,RT,CV,CA} and no model builders are known for CK+{ID,CS,AC,RT,CV,CA}. The presented methodology offers theorem provers and model finders for above mentioned logics. Moreover, the HOL embedding permits use to reason about meta-theorems in an automated way, and naturally extends to First-Order CLs.

Future work includes the systematic analysis of further properties of conditional logics. For example, following [5] and motivated by the results for Problem 5, the systematic verification (respectively exploration) of inclusion and equivalence relations between different conditional logics should be feasible. We also plan to create a library of meaningful and challenging benchmark problems for conditional logics and to evaluate the scalability of our approach. Moreover, a comparison with direct theorem provers for conditional logics and also with related techniques based on translations into first-order logic is needed. However, it is not obvious how these approaches could possibly be applied for reasoning *about* properties of conditional logics as studied in this paper.

Another line of future research is to extend second-order quantifier eliminations techniques like SCAN or SCHEMA to deal with CLs. In fact, both algorithms are not directly suited for reasoning under CLs due to the peculiarities of selection-function semantics.

References

1. P. B. Andrews. General models and extensionality. *J. of Symbol Logic*, 37:395–397, 1972.
2. P. B. Andrews. Church's type theory. In *The Stanford Encyclopedia of Philosophy*. 2009.
3. P. B. Andrews, M. Bishop, S. Issar, D. Nesmith, F. Pfenning, and H. Xi. TPS: A Theorem-Proving System for Classical Type Theory. *J. of Automated Reasoning*, 16(3):321–353, 1996.
4. J. Backes and C. E. Brown. Analytic tableaux for higher-order logic with choice. In *Proc. IJCAR 2010*, volume 6173 of *LNAI*, pages 76–90. Springer, 2010.
5. C. Benzmüller. Verifying the modal logic cube is an easy task (for higher-order automated reasoners). In *Verification, Induction, Termination Analysis – Festschrift C. Walther*, volume 6463 of *LNCS*, pages 117–128. Springer, 2010.
6. C. Benzmüller and L. C. Paulson. Multimodal and intuitionistic logics in simple type theory. *Logic J. of the IGPL*, 18:881–892, 2010.
7. 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 *LNAI*, pages 162–170, 2008.
8. J. C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *Proc. ITP 2010*, volume 6172 of *LNCS*, pages 131–146. Springer, 2010.
9. C. Boutilier. Conditional logics of normality: A modal approach. *Artif. Intell.*, 68(1):87–154, 1994.
10. B.F. Chellas. *Modal Logic: An Introduction*. Cambridge: Cambridge University Press, 1980.
11. A. Church. A formulation of the simple theory of types. *J. of Symbolic Logic*, 5:56–68, 1940.
12. G. Crocco and P. Lamarre. On the connection between non-monotonic inference systems and conditional logics. In *KR*, pages 565–571, 1992.
13. V. Genovese, L. Giordano, V. Gliozzi, and G.L. Pozzato. A constructive conditional logic for access control: a preliminary report. In *Proc. of ECAI 2010*, pages 1073–1074, 2010.
14. T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in Lecture Notes in Computer Science. Springer, 2002.
15. D. Nute. *Topics in conditional logic*. Reidel, Dordrecht, 1980.
16. N. Olivetti and G. L. Pozzato. Theorem proving for conditional logics: CondLean and GoalDuck. *J. of Applied Non-Classical Logics (JANCL)*, 18(4):427–473, 2008.
17. N. Olivetti, G.L. Pozzato, and C. Schwind. A sequent calculus and a theorem prover for standard conditional logics. *ACM Trans. Comput. Log.*, 8(4), 2007.
18. G. L. Pozzato. *Conditional and Preferential Logics: Proof Methods and Theorem Proving*, volume 208 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2010.
19. L. Schröder, D. Pattinson, and D. Hausmann. Optimal tableaux for conditional logics with cautious monotonicity. In *Proc. of ECAI 2010*, pages 707–712, 2010.
20. R. Stalnaker. A theory of conditionals. In N. Rescher, editor, *Studies in Logical Theory, American Philosophical Quarterly, Monograph Series no.2*, page 98112. Blackwell, Oxford, 1968.

21. G. Sutcliffe. The tptp problem library and associated infrastructure. *J. Autom. Reasoning*, 43(4):337–362, 2009.

22. G. Sutcliffe and C. Benzmüller. Automated reasoning in higher-order logic using the TPTP THF infrastructure. *J. of Formalized Reasoning*, 3(1):1–27, 2010.

23. Thomas Raths und Jens Otten. Implementing and evaluating theorem provers for first-order modal logics. In Martin Giese, editor, *Proceedings of FTP 2011 - International Workshop on First-Order Theorem Proving, Bern, Switzerland, July 4, 2011*. CEUR Workshop Proceedings, 2011. See also the online results at `http://www.cs.uni-potsdam.de/ti/iltp/qmltp/download/QMLTP-v1.0-comparison.txt`.

24. T. Weber. *SAT-Based Finite Model Generation for Higher-Order Logic*. Ph.D. thesis, Dept. of Informatics, T.U. München, 2008.