Quantified Conditional Logics are Fragments of HOL

Christoph Benzmüller
Freie Universität Berlin
c.benzmueller@googlemail.com

Valerio Genovese
University of Luxembourg
genovese@di.unito.it

1 Introduction

A semantic embedding of propositional conditional logic in classical higher-order logic HOL (Church's type theory) has been presented in [3]. This embedding exploits the natural correspondence between selection function semantics for conditional logics [10] 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.

In this paper we extend the embedding in [3] to also include quantification over propositions and individuals. This embedding of quantified conditional logic in HOL is sound and complete.

2 Quantified Conditional Logics

We extend propositional conditional logics with quantification over propositional variables and over individuals of a first-order domain. Below, we only consider *constant domains*, i.e., every possible world has the same domain.

Let $\mathscr{IV}$ be a set of first-order (individual) variables, $\mathscr{PV}$ a set of propositional variables, and $\mathscr{SYM}$ a set of predicate symbols of any arity. Formulas of quantified conditional logic are given by the following grammar (where $X^i \in \mathscr{IV}, P \in \mathscr{PV}, k \in \mathscr{SYM}$):

$$\varphi, \psi ::= P \mid k(X^1, \ldots, X^n) \mid \neg\varphi \mid \varphi \vee \psi \mid \forall X.\varphi \mid \forall P.\varphi \mid \varphi \Rightarrow \psi$$

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

Regarding semantics, many different formalizations have been proposed (see [8]), here we focus on the *selection function semantics* [6], which is based on possible world structures and has been successfully used in [9] to develop proof methods for some conditional logics. We adapt selection function semantics for quantified conditional logics.

An *interpretation* is a structure $\mathscr{M} = \langle S, f, D, Q, I\rangle$ where, S is a set of possible items called worlds, $f : S \times 2^S \mapsto 2^S$ is the selection function, D is a non-empty set of *individuals* (the first-order domain), Q is a non-empty collection of subsets of W (the propositional domain), and I is a classical interpretation function where for each n-ary predicate symbol k, $I(k, w) \subseteq D^n$.

A *variable assignment* $g = (g^{iv}, g^{pv})$ is a pair of maps where, $g^{iv} : \mathscr{IV} \mapsto D$ maps each individual variable in $\mathscr{IV}$ to an object in D, and g^{pv} : maps each propositional variable in $\mathscr{PV}$ to a set of worlds in Q.

Satisfiability of a formula φ for an interpretation $\mathscr{M} = \langle S, f, D, Q, I\rangle$, a world $s \in S$, and a variable assignment $g = (g^{iv}, g^{pv})$ is denoted as $M, g, s \models \varphi$ and defined as follows, where $[a/Z]g$ denote the assignment identical to g except that $([a/Z]g)(Z) = a$:

$M, g, s \models k(X^1, \ldots, X^n)$ if and only if $\langle g^{iv}(X^1), \ldots, g^{iv}(X^n)\rangle \in I(k, w)$
$M, g, s \models P$ if and only if $s \in g^{pv}(P)$

$M,g,s \models \neg\varphi$ if and only if $M,g,s \not\models \varphi$ (that is, not $M,g,s \models \varphi$)
$M,g,s \models \varphi \vee \psi$ if and only if $M,g,s \models \varphi$ or $M,g,s \models \psi$
$M,g,s \models \forall X.\varphi$ if and only if $M,([d/X]g^{iv},g^{pv}),s \models \varphi$ for all $d \in D$
$M,g,s \models \forall P.\varphi$ if and only if $M,(g^{iv},[p/P]g^{pv}),s \models \varphi$ for all $p \in Q$
$M,g,s \models \varphi \Rightarrow \psi$ if and only if $M,g,t \models \psi$ for all $t \in S$ such that $t \in f(s,[\varphi])$ where $[\varphi] = \{u \mid M,g,u \models \varphi\}$

An interpretation $\mathscr{M} = \langle S,f,D,Q,I\rangle$ is a *model* if for every variable assignment g and every formula φ, the set of worlds $\{s \in S \mid M,g,s \models \varphi\}$ is a member of Q. As usual, a conditional formula φ is *valid in a model* $\mathscr{M} = \langle S,f,D,Q,I\rangle$, denoted with $\mathscr{M} \models \varphi$, if and only if for all worlds $s \in S$ and variable assignments g holds $\mathscr{M},g,s \models \varphi$. A formula φ is a *valid*, denoted $\models \varphi$, if and only if it is valid in every model.

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

$$\frac{(\varphi_1 \wedge \ldots \wedge \varphi_n) \leftrightarrow \psi}{(\varphi_0 \Rightarrow \varphi_1 \wedge \ldots \wedge \varphi_0 \Rightarrow \varphi_n) \rightarrow (\varphi_0 \Rightarrow \psi)}\ (RCK) \qquad \frac{\varphi \leftrightarrow \varphi'}{(\psi \Rightarrow \varphi) \leftrightarrow (\psi \Rightarrow \varphi')}\ (RCEC)$$

We refer to *CK* [6] as the minimal quantified conditional logic closed under rules RCEA, RCEC and RCK. In what follows, only quantified conditional logics extending CK are considered.

3 Classical Higher-Order Logic

HOL is a logic based on simply typed λ-calculus [7, 2]. The set $\mathscr{T}$ of simple types in HOL is usually freely generated from a set of basic types $\{o,i\}$ using the function type constructor $\rightarrow$. Here we instead consider a set of basic type $\{o,i,u\}$, where o denotes the type of Booleans, and where i and u denote some non-empty domains. Without loss of generality, we will later identify i with a set of worlds and u with a domain of individuals.

Let $\alpha,\beta,o \in \mathscr{T}$. The *terms* of HOL are defined by the grammar (p_α denotes typed constants and X_α typed variables distinct from p_α):

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

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: for example, $\wedge$ 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].

A *frame* is a collection $\{D_\alpha\}_{\alpha\in\mathscr{T}}$ of nonempty sets called *domains* such that $D_o = \{T,F\}$ where T represents truth and F falsehood, $D_i \neq \emptyset$ and $D_u \neq \emptyset$ are chosen arbitrary, and $D_{\alpha\rightarrow\beta}$ are collections of total functions mapping D_α into D_β.

An *interpretation* is a tuple $\langle\{D_\alpha\}_{\alpha\in\mathscr{T}},I\rangle$ where $\{D_\alpha\}_{\alpha\in\mathscr{T}}$ 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 $\mathscr{V}$ such that $\mathscr{V}(\phi,s_\alpha)\in D_\alpha$ 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 $\alpha,\beta\in\mathscr{T}$): $\mathscr{V}(\phi,X_\alpha)=\phi(X_\alpha)$, $\mathscr{V}(\phi,p_\alpha)=I(p_\alpha)$, $\mathscr{V}(\phi,(l_{\alpha\to\beta}\ r_\alpha))=(\mathscr{V}(\phi,l_{\alpha\to\beta}))(\mathscr{V}(\phi,r_\alpha))$, and $\mathscr{V}(\phi,\lambda X_\alpha.s_\beta)$ represents the function from D_α into D_β whose value for each argument $z\in D_\alpha$ is $\mathscr{V}(\phi[z/X_\alpha],s_\beta)$, 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 $\mathscr{H}=\langle\{D_\alpha\}_{\alpha\in\mathscr{T}},I\rangle$ is an *Henkin model* the function $\mathscr{V}$ is uniquely determined and $\mathscr{V}(\phi,s_\alpha)\in D_\alpha$ is called the denotation of s_α. $\mathscr{H}$ 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 $\mathscr{H}$ if and only if $\mathscr{V}(\phi,A)=T$ for all variable assignments ϕ. In this case we write $\mathscr{H}\models A$. A is (Henkin) valid, denoted as $\models A$, if and only if $\mathscr{H}\models A$ for all Henkin models $\mathscr{H}$.

PROP. 3.1. *Let $\mathscr{V}$ be the valuation function of Henkin model $\mathscr{H}$. The following properties hold for all assignments ϕ, terms $s_o,t_o,l_\alpha,r_\alpha$, and variables X_α,V_α (for $\alpha\in\mathscr{T}$): $\mathscr{V}(\phi,(\neg s_o))=T$ if and only if $\mathscr{V}(\phi,s_o)=F$, $\mathscr{V}(\phi,(s_o\vee t_o))=T$ if and only if $\mathscr{V}(\phi,s_o)=T$ or $\mathscr{V}(\phi,s_o)=T$, $\mathscr{V}(\phi,(s_o\wedge t_o))=T$ if and only if $\mathscr{V}(\phi,s_o)=T$ and $\mathscr{V}(\phi,s_o)=T$, $\mathscr{V}(\phi,(s_o\to t_o))=T$ if and only if $\mathscr{V}(\phi,s_o)=F$ or $\mathscr{V}(\phi,s_o)=T$, $\mathscr{V}(\phi,(\forall X_\alpha.s_o))=\mathscr{V}(\phi,(\Pi_{(\alpha\to o)\to o}\ (\lambda X_\alpha.s_o)))=T$ if and only if for all $v\in D_\alpha$ holds $\mathscr{V}(\phi[v/V_\alpha],((\lambda X_\alpha.s_o)\ V))=T$, and if $l_\alpha=_{\beta\eta}r_\alpha$ then $\mathscr{V}(\phi,l_\alpha)=\mathscr{V}(\phi,r_\alpha)$*

4 Embedding Quantified Conditional Logics in HOL

Quantified 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 worlds.

DEF. 4.1. *The mapping $\lfloor\cdot\rfloor$ translates formulas φ of quantified conditional logic CK into HOL terms $\lfloor\varphi\rfloor$ of type $i\to o$. The mapping is recursively defined as follows:*

$$\begin{array}{lcl@{\qquad}lcl}
\lfloor P\rfloor & = & P_{i\to o} & \lfloor\varphi\vee\psi\rfloor & = & \vee_{(i\to o)\to(i\to o)\to(i\to o)}\ \lfloor\varphi\rfloor\lfloor\psi\rfloor\\
\lfloor k(X^1,\dots,X^n)\rfloor & = & (\lfloor k\rfloor\lfloor X^1\rfloor\dots\lfloor X^n\rfloor)\rfloor & \lfloor\varphi\Rightarrow\psi\rfloor & = & \Rightarrow_{(i\to o)\to(i\to o)\to(i\to o)}\ \lfloor\varphi\rfloor\lfloor\psi\rfloor\\
 & = & (k_{u^n\to(i\to o)}\ X^1_u\dots X^n_u) & \lfloor\forall X.\varphi\rfloor & = & \Pi_{(u\to(i\to o))\to(i\to o)}\ \lambda X_u.\lfloor\varphi\rfloor\\
\lfloor\neg\varphi\rfloor & = & \neg_{i\to o}\ \lfloor\varphi\rfloor & \lfloor\forall P.\varphi\rfloor & = & \Pi_{((i\to o)\to(i\to o))\to(i\to o)}\ \lambda P_{i\to o}.\lfloor\varphi\rfloor
\end{array}$$

$P_{i\to o}$ and $X^1_u,\dots,X^n_u$ are HOL variables and $k_{u^n\to(i\to o)}$ is a HOL constant. $\neg_{i\to o}$, $\vee_{(i\to o)\to(i\to o)\to(i\to o)}$, $\Rightarrow_{(i\to o)\to(i\to o)\to(i\to o)}$, $\Pi_{(u\to(i\to o))\to(i\to o)}$ and $\Pi_{((i\to o)\to(i\to o))\to(i\to o)}$ realize the quantified conditional logics connectives in HOL. They abbreviate the following proper HOL terms:

$$\begin{array}{ll}
\neg_{(i\to o)\to(i\to o)} & =\lambda A_{i\to o}.\lambda X_i.\neg(A\,X)\\
\vee_{(i\to o)\to(i\to o)\to(i\to o)} & =\lambda A_{i\to o}.\lambda B_{i\to o}.\lambda X_i.(A\,X)\vee(B\,X)\\
\Rightarrow_{(i\to o)\to(i\to o)\to(i\to o)} & =\lambda A_{i\to o}.\lambda B_{i\to o}.\lambda X_i.\forall W_i.(f\,X\,A\,W)\to(B\,W)\\
\Pi_{(u\to(i\to o))\to(i\to o)} & =\lambda Q_{u\to(i\to o)}.\lambda W_i.\forall X_u.(Q\,X\,W)\\
\Pi_{((i\to o)\to(i\to o))\to(i\to o)} & =\lambda R_{(i\to o)\to(i\to o)}.\lambda W_i.\forall P_{i\to o}.(R\,P\,W)
\end{array}$$

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).

This mapping induces mappings $\lfloor\mathscr{IV}\rfloor$, $\lfloor\mathscr{PV}\rfloor$ and $\lfloor\mathscr{SYM}\rfloor$ of the sets $\mathscr{IV}$, $\mathscr{PV}$ and $\mathscr{SYM}$ respectively.

Analyzing the validity of a translated formula $\lfloor \varphi \rfloor$ for a world represented by term t_i corresponds to evaluating the application $(\lfloor \varphi \rfloor\ t_i)$. In line with [4], we define $\text{vld}_{(i\to o)\to o} = \lambda A_{i\to o}.\forall S_i.(A\ S)$. With this definition, validity of a quantified conditional formula φ in CK corresponds to the validity of the corresponding formula $(\text{vld}\ \lfloor \varphi \rfloor)$ in HOL, and vice versa.

5 Soundness and Completeness

To prove the soundness and completeness of the embedding, a mapping from selection function models into Henkin models is employed. This mapping will employ a corresponding mapping of variable assignments for quantified conditional logics into variable assignments for HOL.

DEF. 5.1 (Mapping of Variable Assignments). *Let $g = (g^{iv} : \mathscr{IV} \longrightarrow D, g^{pv} : \mathscr{PV} \longrightarrow Q)$ be a variable assignment for a quantified conditional logic. We define the corresponding variable assignment $\lfloor g \rfloor = (\lfloor g^{iv} \rfloor : \lfloor \mathscr{IV} \rfloor \longrightarrow D, \lfloor g^{pv} \rfloor : \lfloor \mathscr{PV} \rfloor \longrightarrow Q)$ for HOL so that $\lfloor g \rfloor(X_u) = \lfloor g \rfloor(\lfloor X \rfloor) = g(X)$ and $\lfloor g \rfloor(P_{i\to o}) = \lfloor g \rfloor(\lfloor P \rfloor) = g(P)$ for all $X_u \in \lfloor \mathscr{IV} \rfloor$ and $P_{i\to o} \in \lfloor \mathscr{PV} \rfloor$. Finally, a variable assignment $\lfloor g \rfloor$ is extended to an assignment for variables Z_α of arbitrary type by choosing $\lfloor g \rfloor(Z_\alpha) = d \in D_\alpha$ arbitrary, if $\alpha \neq u, i \to o$.*

DEF. 5.2 (Henkin model $\mathscr{H}^{\mathscr{M}}$). *Given a quantified conditional logic model $\mathscr{M} = \langle S, f, D, Q, I \rangle$. The Henkin model $\mathscr{H}^{\mathscr{M}} = \langle \{D_\alpha\}_{\alpha \in \mathscr{T}}, I \rangle$ for $\mathscr{M}$ is defined as follows: D_i is chosen as the set of possible worlds S, D_u is chosen as the first-order domain D (cf. definition of $\lfloor g^{iv} \rfloor$), $D_{i\to o}$ is chosen as the set of sets of possible worlds Q (cf. definition of $\lfloor g^{pv} \rfloor$)[1], and all other sets $D_{\alpha\to\beta}$ are chosen as (not necessarily full) sets of functions from D_α to D_β. For all sets $D_{\alpha\to\beta}$ the rule that everything denotes must be obeyed, in particular, we require that the sets $D_{u^n\to(i\to o)}$ and $D_{i\to(i\to o)\to(i\to o)}$ contain the elements $Ik_{u^n\to(i\to o)}$ and $If_{i\to(i\to o)\to(i\to o)}$ as characterized below.*

The interpretation I is constructed as follows: (i) Let $k_{u^n\to(i\to o)} = \lfloor k \rfloor$ for n-ary $k \in \mathscr{SYM}$ and let $X^i_u = \lfloor X^i \rfloor$ for $X^i \in \mathscr{IV}$, $i = 1,\ldots,n$. We choose $Ik_{u^n\to(i\to o)} \in D_{u^n\to(i\to o)}$ such that $(I k_{u^n\to(i\to o)})(\lfloor g \rfloor(X^1_u),\ldots,\lfloor g \rfloor(X^n_u), w) = T$ for all worlds $w \in D_i$ such that $\mathscr{M}, g, w \models k(X^1,\ldots,X^n)$, that is, if $\langle g^{iv}(X^1),\ldots,g^{iv}(X^n)\rangle \in I(k,w)$. Otherwise we choose $(I k_{u^n\to(i\to o)})(\lfloor g \rfloor(X^1_u),\ldots,\lfloor g \rfloor(X^n_u), w) = F$. (ii) We choose $If_{i\to(i\to o)\to(i\to o)} \in D_{i\to(i\to o)\to(i\to o)}$ such that $(If_{i\to(i\to o)\to(i\to o)})(s,q,t) = T$ for all worlds $s,t \in D_i$ and $q \in D_{i\to o}$ with $t \in f(s, \{x \in S \mid q(x) = T\})$ in $\mathscr{M}$. Otherwise we choose $(If_{i\to(i\to o)\to(i\to o)})(s,q,t) = F$. (iii) For all other constants s_α, choose Is_α arbitrary.[2]

It is not hard to verify that $\mathscr{H}^{\mathscr{M}}$ is a Henkin model.

LEMMA 5.3. *Let $\mathscr{H}^{\mathscr{M}}$ be a Henkin model for a selection function model $\mathscr{M}$. For all quantified conditional logic formulas δ, variable assignments g and worlds s it holds: $\mathscr{M}, g, s \models \delta$ if and only if $\mathscr{V}(\lfloor g \rfloor[s/S_i], (\lfloor \delta \rfloor\ S_i)) = T$*

Proof. The proof is by induction on the structure of δ. The cases for $\delta = P$, $\delta = k(X^1,\ldots,X^n)$, $\delta = (\neg\varphi)$, $\delta = (\varphi \vee \psi)$, and $\delta = (\varphi \Rightarrow \psi)$ are similar to Lemma 1 in [3]. The cases for $\delta = \forall X.\varphi$ and $\delta = \forall P.\varphi$ adapt the respective cases from Lemmas 4.3 and 4.7 in [5].

□

THEOREM 5.4 (Soundness and Completeness). $\models (\mathit{vld}\ \lfloor \varphi \rfloor)$ *in HOL if and only if* $\models \varphi$ *in CK*

[1] To keep things simple, we identify sets with their characteristic functions.

[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 $, k_{u^n\to(i\to o)}$, and the logical connectives.

Proof. (Soundness) The proof is by contraposition. Assume $\not\models \varphi$ in CK, that is, there is a model $\mathscr{M} = \langle S, f, D, Q, I\rangle$, a variable assignment g and a world $s \in S$, such that $\mathscr{M}, g, s \not\models \varphi$. By Lemma 5.3 we have that $\mathscr{V}(\lfloor g\rfloor[s/S_i], (\lfloor\varphi\rfloor\ S)) = F$ in Henkin model $\mathscr{H}^{\mathscr{M}} = \langle\{D_\alpha\}_{\alpha\in\mathscr{T}}, I\rangle$ for $\mathscr{M}$. Thus, by Prop. 3.1, definition of vld and since $(\forall S_i.\lfloor\varphi\rfloor\ S) =_{\beta\eta} (\text{vld}\ \lfloor\varphi\rfloor)$ we know that $\mathscr{V}(\lfloor g\rfloor, (\forall S_i.\lfloor\varphi\rfloor\ S)) = \mathscr{V}(\lfloor g\rfloor, (\text{vld}\ \lfloor\varphi\rfloor)) = F$. Hence, $\mathscr{H}^{\mathscr{M}} \not\models (\text{vld}\ \lfloor\varphi\rfloor)$, and thus $\not\models (\text{vld}\ \lfloor\varphi\rfloor)$ in HOL.

(Completeness) The proof is again by contraposition. Assume $\not\models (\text{vld}\ \lfloor\varphi\rfloor)$ in HOL, that is, there is a Henkin model $\mathscr{H} = \langle\{D_\alpha\}_{\alpha\in\mathscr{T}}, I\rangle$ and a variable assignment ϕ with $\mathscr{V}(\phi, (\text{vld}\ \lfloor\varphi\rfloor)) = F$. Without loss of generality we can assume that Henkin Model $\mathscr{H}$ is in fact a Henkin model $\mathscr{H}^{\mathscr{M}}$ for a corresponding quantified conditional logic model $\mathscr{M}$ and that $\Phi = \lfloor g\rfloor$ for a corresponding quantified conditional logic variable assignment g. By Prop. 3.1 and since $(\text{vld}\ \lfloor\varphi\rfloor) =_{\beta\eta} (\forall S_i.\lfloor\varphi\rfloor\ S)$ we have $\mathscr{V}(\lfloor g\rfloor, (\forall S_i.\lfloor\varphi\rfloor\ S)) = F$, and hence, by definition of vld, $\mathscr{V}(\lfloor g\rfloor[s/S_i], \lfloor\varphi\rfloor\ S) = F$ for some $s \in D$. By Lemma 5.3 we thus know that $\mathscr{M}, g, s \not\models \varphi$, and hence $\not\models \varphi$ in CK. □

6 Conclusion

We have presented an embedding of quantified conditional logics in HOL. This embedding enables the uniform application of higher-order automated theorem provers and model finders for reasoning about and within quantified conditional logics. In previous work we have studied related embeddings in HOL, including propositional conditional logics [3] and quantified multimodal logics [5]. First experiments with these embeddings have provided evidence for their practical relevance. Moreover, an independent case study on reasoning in quantified modal logics shows that the embeddings based approach may even outperform specialist reasoners quantified modal logics [12]. Future work will investigate whether HOL reasoners perform similarly well also for quantified conditional logics. For a first impression of such studies we refer to the Appendices A and B, where we also present the concrete encoding of our embedding in TPTP THF0 [11] syntax. Unfortunately we are not aware of any other (direct or indirect) prover for quantified conditional logics that could be used for comparison.

References

[1] P. B. Andrews. General models and extensionality. *J. of Symbolic Logic*, 37:395–397, 1972.

[2] P. B. Andrews. Church's type theory. In *The Stanford Encyclopedia of Philosophy*. 2009.

[3] C. Benzmüller, D. Gabbay, V. Genovese, and D. Rispoli. Embedding and automating conditional logics in classical higher-order logic. Technical report, 2011. http://arxiv.org/abs/1106.3685.

[4] C. Benzmüller and L.C. Paulson. Multimodal and intuitionistic logics in simple type theory. *Logic J. of the IGPL*, 18:881–892, 2010.

[5] C. Benzmüller and L.C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue om Multimodal Logics)*, 2011. To appear. See also http://arxiv.org/abs/0905.2435.

[6] B.F. Chellas. *Modal Logic: An Introduction*. Cambridge: Cambridge University Press, 1980.

[7] A. Church. A formulation of the simple theory of types. *J. of Symbolic Logic*, 5:56–68, 1940.

[8] D. Nute. *Topics in conditional logic*. Reidel, Dordrecht, 1980.

[9] 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.

[10] R. Stalnaker. A theory of conditionals. In N. Rescher, editor, *Studies in Logical Theory, American Philosophical Quarterly, Monograph Series no.2*, page 98–112. Blackwell, Oxford, 1968.

[11] 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.

[12] T. Raths und J. Otten. Implementing and evaluating theorem provers for first-order modal logics. In M. Giese, editor, *Proceedings of FTP 2011*. CEUR Workshop Proceedings, 2011.

A The Embedding of Quantified Conditional Logic in HOL in THF0 Syntax

We present an encoding of our embedding of quantified conditional logics in HOL in the TPTP THF0 [11] syntax.

Satisfiability of this embedding is shown by the HOL reasoner Satallax[3] in only 0.01 seconds.

```
%------------------------------------------------------------------------
%---- reserved constant for selection function f
thf(f_type,type,(
    f: $i > ( $i > $o ) > $i > $o )).

%---- 'not' in conditional logic
thf(cnot_type,type,(
    cnot: ( $i > $o ) > $i > $o )).

thf(cnot_def,definition,
    ( cnot
    = ( ^ [Phi: $i > $o,X: $i] :
          ~ ( Phi @ X ) ) )).

%---- 'or' in conditional logic
thf(cor_type,type,(
    cor: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(cor_def,definition,
    ( cor
    = ( ^ [Phi: $i > $o,Psi: $i > $o,X: $i] :
          ( ( Phi @ X )
          | ( Psi @ X ) ) ) )).

%---- 'true' in conditional logic
thf(ctrue_type,type,(
    ctrue: $i > $o )).

thf(ctrue_def,definition,
    ( ctrue
    = ( ^ [X: $i] : $true ) )).

%---- 'false' in conditional logic
thf(cfalse_type,type,(
    cfalse: $i > $o )).

thf(cfalse_def,definition,
    ( cfalse
    = ( ^ [X: $i] : $false ) )).

%---- 'conditional implication' in conditional logic
thf(ccond_type,type,(
    ccond: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(ccond_def,definition,
```

[3]http://www.ps.uni-saarland.de/~cebrown/satallax/

```
    ( ccond
    = ( ^ [Phi: $i > $o,Psi: $i > $o,X: $i] :
        ! [W: $i] :
          ( ( f @ X @ Phi @ W )
         => ( Psi @ W ) ) ) )).

%---- 'and' in conditional logic
thf(cand_type,type,(
    cand: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(cand_def,definition,
    ( cand
    = ( ^ [Phi: $i > $o,Psi: $i > $o,X: $i] :
          ( ( Phi @ X )
          & ( Psi @ X ) ) ) )).

%---- 'conditional equivalence' in conditional logic
thf(ccondequiv_type,type,(
    ccondequiv: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(ccondequiv_def,definition,
    ( ccondequiv
    = ( ^ [Phi: $i > $o,Psi: $i > $o] :
          ( cand @ ( ccond @ Phi @ Psi ) @ ( ccond @ Psi @ Phi ) ) ) )).

%---- 'material implication' in conditional logic
thf(cimpl_type,type,(
    cimpl: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(cimpl_def,definition,
    ( cimpl
    = ( ^ [Phi: $i > $o,Psi: $i > $o,X: $i] :
          ( ( Phi @ X )
         => ( Psi @ X ) ) ) )).

%---- 'material equivalence' in conditional logic
thf(cequiv_type,type,(
    cequiv: ( $i > $o ) > ( $i > $o ) > $i > $o )).

thf(cequiv_def,definition,
    ( cequiv
    = ( ^ [Phi: $i > $o,Psi: $i > $o] :
          ( cand @ ( cimpl @ Phi @ Psi ) @ ( cimpl @ Psi @ Phi ) ) ) )).

%---- 'universal quantification (individuals)' in conditional logic
thf(cforall_ind_type,type,(
    cforall_ind: ( mu > $i > $o ) > $i > $o )).

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

```
%---- 'universal quantification (propositions)' in conditional logic
thf(cforall_prop_type,type,(
    cforall_prop: ( ( $i > $o ) > $i > $o ) > $i > $o )).

thf(cforall_prop,definition,
    ( cforall_prop
    = ( ^ [Phi: ( $i > $o ) > $i > $o,W: $i] :
        ! [P: $i > $o] :
          ( Phi @ P @ W ) ) )).

%---- 'existential quantification (individuals)' in conditional logic
thf(cexists_ind_type,type,(
    cexists_ind: ( mu > $i > $o ) > $i > $o )).

thf(cexists_ind,definition,
    ( cexists_ind
    = ( ^ [Phi: mu > $i > $o] :
          ( cnot
          @ ( cforall_ind
            @ ^ [X: mu] :
                ( cnot @ ( Phi @ X ) ) ) ) ) )).

%---- 'existential quantification (propositions)' in conditional logic
thf(cexists_prop_type,type,(
    cexists_prop: ( ( $i > $o ) > $i > $o ) > $i > $o )).

thf(cexists_prop,definition,
    ( cexists_prop
    = ( ^ [Phi: ( $i > $o ) > $i > $o] :
          ( cnot
          @ ( cforall_prop
            @ ^ [P: $i > $o] :
                ( cnot @ ( Phi @ P ) ) ) ) ) )).

%---- 'validity' of a conditional logic formula
thf(valid_type,type,(
    valid: ( $i > $o ) > $o )).

thf(valid_def,definition,
    ( valid
    = ( ^ [Phi: $i > $o] :
        ! [S: $i] :
          ( Phi @ S ) ) )).
%----------------------------------------------------------------------
```

B The Barcan Formula and the Converse Barcan Formula

Using the above THF0 encoding, the Barcan formula $(\forall X.A \Rightarrow B(x)) \rightarrow (A \Rightarrow \forall X.B(x))$ can be encoded in THF0 as given below. The HOL provers LEO-II[4] and Satallax can both prove this theorem in 0.01 seconds. This confirms that we our encoding assumes constant domain structure.

```
%------------------------------------------------------------------------
include('CK_axioms.ax').

%---- conjecture statement
thf(a,type,(
    a: $i > $o )).

thf(b,type,(
    b: mu > $i > $o )).

thf(bf,conjecture,
    ( valid
    @ ( cimpl
      @ ( cforall_ind
        @ ^ [X: mu] :
            ( ccond @ a @ ( b @ X ) ) )
      @ ( ccond @ a
        @ ( cforall_ind
          @ ^ [X: mu] :
              ( b @ X ) ) ) ) )).
%------------------------------------------------------------------------
```

The converse Barcan formula $(A \Rightarrow \forall X.B(x)) \rightarrow (\forall X.A \Rightarrow B(x))$ can be encoded analogously. Again, the HOL provers LEO-II and Stallax need only 0.01 seconds to prove this theorem.

```
%------------------------------------------------------------------------
include('CK_axioms.ax').

%---- conjecture statement
thf(a,type,(
    a: $i > $o )).

thf(b,type,(
    b: mu > $i > $o )).

thf(bf,conjecture,
    ( valid
    @ ( cimpl
      @ ( cforall_ind
        @ ^ [X: mu] :
            ( ccond @ a @ ( b @ X ) ) )
      @ ( ccond @ a
        @ ( cforall_ind
          @ ^ [X: mu] :
              ( b @ X ) ) ) ) )).
%------------------------------------------------------------------------
```

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