Quantified Conditional Logics are Fragments of HOL

Christoph Benzmüller  
Free University Berlin  
c.benzmueller@googlemail.com

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

Abstract

A semantic embedding of quantified conditional logic in classical higher-order logic is presented.

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 $\mathcal{W}$ be a set of first-order (individual) variables, $\mathcal{P}$ a set of propositional variables, and $\mathcal{M}$ a set of predicate symbols of any arity. Formulas of quantified conditional logic are given by the following grammar (where $X_i \in \mathcal{W}$, $P \in \mathcal{P}$, $k \in \mathcal{M}$):

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

From the selected set of primitive connectives, other logical connectives can be introduced as abbreviations: for example, $\varphi \land \psi$, $\varphi \Rightarrow \psi$ (material implication), and $\exists X. \varphi$ abbreviate $\neg (\varphi \lor \neg \psi)$, $\neg \varphi \lor \psi$ 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 $\mathcal{M} = \langle S, f, D, Q, I \rangle$ where, $S$ is a set of possible items called worlds, $f : S \times 2^S \rightarrow 2^S$ 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 $k$, $I(k, w) \subseteq D^n$.

*This work has been presented at the conference on Non-classical Modal and Predicate Logics 2011, Guangzhou (Canton), China, 5-9 December 2011.
A variable assignment $g = (g^i, g^\mu)$ is a pair of maps where, $g^i : \mathcal{I} \ni \Phi \mapsto D$ maps each individual variable in $\mathcal{I}$ to an object in $D$, and $g^\mu : \text{maps each propositional variable in } \mathcal{P} \text{ to a set of worlds in } Q$.

Satisfaction of a formula $\varphi$ for an interpretation $\mathcal{M} = (\mathcal{S}, f, D, Q, I)$, a world $s \in \mathcal{S}$, and a variable assignment $g = (g^i, g^\mu)$ 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 $(g^i(X^1), \ldots, g^i(X^n)) \in I(k, w)$
- $M, g, s \models P$ if and only if $s \in g^\mu(P)$
- $M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$ (that is, not $M, g, s \models \varphi$)
- $M, g, s \models \varphi \lor \psi$ 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 $M, ([d/X]g^i, g^\mu), s \models \varphi$ for all $d \in D$
- $M, g, s \models \forall P. \varphi$ if and only if $M, (g^i, [p/P]g^\mu), s \models \varphi$ 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 $\mathcal{M} = (\mathcal{S}, f, D, Q, I)$ is a model if for every variable assignment $g$ and every formula $\varphi$, the set of worlds $\{s \in \mathcal{S} \mid M, g, s \models \varphi\}$ is a member of $Q$. As usual, a conditional formula $\varphi$ is valid in a model $\mathcal{M} = (\mathcal{S}, f, D, Q, I)$, denoted with $\mathcal{M} \models \varphi$, if and only if for all worlds $s \in \mathcal{S}$ and variable assignments $g$ holds $\mathcal{M}, g, s \models \varphi$. A formula $\varphi$ 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 $\varphi$ w.r.t. a given model $\mathcal{M}$) instead of $\varphi$. This approach has the consequence of forcing the so-called normality property: given a model $\mathcal{M}$, if $\varphi$ and $\varphi'$ 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)

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

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

$$\frac{(\varphi_1 \land \ldots \land \varphi_n) \leftrightarrow \psi}{(\varphi_0 \Rightarrow \varphi_1 \land \ldots \land \varphi_n \Rightarrow \varphi_0) \Rightarrow (\varphi_0 \Rightarrow \psi)} \quad \text{(RCK)} \quad \text{and} \quad \frac{\varphi \leftrightarrow \varphi'}{(\psi \Rightarrow \varphi) \leftrightarrow (\psi \Rightarrow \varphi')} \quad \text{(RCEC)}$$

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 $\lambda$-calculus [7][2]. The set $\mathcal{T}$ of simple types in HOL is usually freely generated from a set of basic types $\{o, i\}$ using the function type constructor $\to$. 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 $\alpha, \beta, o \in \mathcal{T}$. The terms of HOL are defined by the grammar ($p_\alpha$ denotes typed constants and $X_\alpha$ typed variables distinct from $p_\alpha$):

$$s, t ::= p_\alpha \mid X_\alpha \mid (\lambda X_\alpha.s_\beta)_{\alpha \to \beta} \mid (s_{\alpha \to \beta})_{\beta} \mid (\neg o \to o)_{o} \mid (s_{o} \lor o \to o)_{o} \mid (\Pi (o \to o) \to o)_{o}$$

Complex typed terms are constructed via abstraction and application. The primitive logical connectives are $\neg o \to o$, $\lor o \to o$ and $\Pi (o \to o) \to o$ (for each type $\alpha$). From these, other logical connectives can
be introduced as abbreviations: for example, \(\land\) and \(\rightarrow\) abbreviate the terms \(\lambda A. \lambda B. \neg(\neg A \land \neg B)\) and \(\lambda A. \lambda B. \neg A \lor B\), etc. HOL terms of type \(o\) are called formulas. Binder notation \(\forall X \alpha. s_\alpha\) is used as an abbreviation for \(\langle\Pi(\alpha \rightarrow o), (\lambda X \alpha.s_\alpha)\rangle\). Substitution of a term \(A_\alpha\) for a variable \(X_\alpha\) in a term \(B_\beta\) 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 \(\{D_\alpha\}_{\alpha \in \mathcal{F}}\) of nonempty sets called domains such that \(D_i = \{T,F\}\) where \(T\) represents truth and \(F\) falsehood, \(D_1 \neq \emptyset\) and \(D_2 \neq \emptyset\) are chosen arbitrary, and \(D_{\alpha \rightarrow \beta}\) are collections of total mappings \(D_\alpha\) into \(D_\beta\).

An interpretation is a tuple \(\langle\{D_\alpha\}_{\alpha \in \mathcal{F}}, I\rangle\) where \(\{D_\alpha\}_{\alpha \in \mathcal{F}}\) is a frame and where function \(I\) maps each typed constant \(c_\alpha\) to an appropriate element of \(D_\alpha\), which is called the denotation of \(c_\alpha\). The denotations of \(\neg, \lor, \land, \Pi(\alpha \rightarrow o)\), \(\forall\), \(\exists\), \(\lambda\), and \(\alpha\) are always chosen as usual. A variable assignment \(\phi\) maps variables \(X_\alpha\) to elements in \(D_\alpha\).

An interpretation is a Henkin model (general model) if and only if there is a binary valuation function \(\forall^\prime\) such that \(\forall^\prime(\phi, s_\alpha) \in D_\alpha\) for each variable assignment \(\phi\) and term \(s_\alpha\), and the following conditions are satisfied for all \(\phi, variables X_\alpha, constants p_\alpha\), and terms \(I_{\alpha \rightarrow \beta}.r_\alpha.s_\beta\) (for \(\alpha, \beta \in \mathcal{F}\)): \(\forall^\prime(\phi, X_\alpha) = \phi(X_\alpha), \forall^\prime(\phi, p_\alpha) = I(p_\alpha), \forall^\prime(\phi, (I_{\alpha \rightarrow \beta}.r_\alpha)) = (\forall^\prime(\phi, I_{\alpha \rightarrow \beta}))(\forall^\prime(\phi, r_\alpha))\), and \(\forall^\prime(\phi, \lambda X_\alpha. s_\beta)\) represents the function from \(D_\alpha\) into \(D_\beta\) whose value for each argument \(z \in D_\alpha\) is \(\forall^\prime(\phi[z/X_\alpha], s_\beta)\), 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 \(\mathcal{H} = \langle\{D_\alpha\}_{\alpha \in \mathcal{F}}, I\rangle\) is an Henkin model the function \(\forall^\prime\) is uniquely determined and \(\forall^\prime(\phi, s_\alpha) \in D_\alpha\) is called the denotation of \(s_\alpha\). \(\mathcal{H}\) is called a standard model if and only if for all \(\alpha\) and \(\beta, D_{\alpha \rightarrow \beta}\) is the set of all functions from \(D_\alpha\) into \(D_\beta\). 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 \(\forall^\prime(\phi, A) = T\) for all variable assignments \(\phi\). 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 \(\mathcal{H}\).

**PROP. 3.1.** Let \(\forall^\prime\) be the valuation function of Henkin model \(\mathcal{H}\). The following properties hold for all assignments \(\phi, terms s_\alpha, t_\alpha, I_{\alpha \rightarrow \beta}.r_\alpha, and variables X_\alpha, V_\alpha\) (for \(\alpha \in \mathcal{F}\)): \(\forall^\prime(\phi, (\neg s_\alpha)) = T\) if and only if \(\forall^\prime(\phi, s_\alpha) = F\), \(\forall^\prime(\phi, (s_\alpha \lor t_\alpha)) = T\) if and only if \(\forall^\prime(\phi, s_\alpha) = T\) or \(\forall^\prime(\phi, t_\alpha) = T\), \(\forall^\prime(\phi, (s_\alpha \land t_\alpha)) = T\) if and only if \(\forall^\prime(\phi, s_\alpha) = T\) and \(\forall^\prime(\phi, t_\alpha) = T\), \(\forall^\prime(\phi, (s_\alpha \rightarrow t_\alpha)) = T\) if and only if \(\forall^\prime(\phi, s_\alpha) = F\) or \(\forall^\prime(\phi, t_\alpha) = T\), \(\forall^\prime(\phi, (\forall X_\alpha. D_{\alpha \rightarrow o}. V_\alpha)) = \forall^\prime(\phi, (\Pi(\alpha \rightarrow o). (\lambda X_\alpha. s_\beta))) = T\) if and only if for all \(v \in D_\alpha\) holds \(\forall^\prime(\phi[v/V_\alpha], (\lambda X_\alpha. s_\beta) V) = T\), and if \(l_\alpha =_{\beta\eta} r_\alpha\) then \(\forall^\prime(\phi, l_\alpha) = \forall^\prime(\phi, r_\alpha)\).

## 4 Embedding Quantified Conditional Logics in HOL

Quantified conditional logic formulas are identified with certain HOL terms (predicates) of type \(i \rightarrow o\). They can be applied to terms of type \(i\), which are assumed to denote possible worlds.

**DEF. 4.1.** The mapping \([\cdot]\) translates formulas \(\varphi\) of quantified conditional logic \(CK\) into HOL terms \([\varphi]\) of type \(i \rightarrow o\). The mapping is recursively defined as follows:

\[
\begin{align*}
[P] & = P_{i \rightarrow o} \\
[k(X^1, \ldots, X^n)] & = ([k][X^1] \ldots [X^n]) \\
[\neg \varphi] & = \neg_{i \rightarrow o}[\varphi] \\
[\varphi \lor \psi] & = \lor_{(i \rightarrow o) \rightarrow (i \rightarrow o)}[\varphi][\psi] \\
[\varphi \rightarrow \psi] & = \rightarrow_{(i \rightarrow o) \rightarrow (i \rightarrow o)}[\varphi][\psi] \\
[\forall X_\alpha \varphi] & = \Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \lambda X_\alpha. [\varphi] \\
[\forall P_\alpha \varphi] & = \Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \lambda P_{i \rightarrow o}. [\varphi]
\end{align*}
\]

\(P_{i \rightarrow o}\) and \(X^1_\alpha, \ldots, X^n_\alpha\) are HOL variables and \(k_{i \rightarrow o}\) is a HOL constant. \(\neg_{i \rightarrow o}[\varphi]\), \(\lor_{(i \rightarrow o) \rightarrow (i \rightarrow o)}[\varphi][\psi]\), \(\rightarrow_{(i \rightarrow o) \rightarrow (i \rightarrow o)}[\varphi][\psi]\), \(\Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \lambda X_\alpha. [\varphi]\), and \(\Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \lambda P_{i \rightarrow o}. [\varphi]\) realize the quantified conditional logics.
connectives in HOL. They abbreviate the following proper HOL terms:

\[ \neg (i \rightarrow o) \quad = \lambda A_{i \rightarrow o} \cdot \lambda X_i \cdot \neg (A X) \]
\[ \lor (i \rightarrow o) \rightarrow (i \rightarrow o) \quad = \lambda A_{i \rightarrow o} \cdot \lambda B_{i \rightarrow o} \cdot \lambda X_i \cdot (A X) \lor (B X) \]
\[ \Rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o) \quad = \lambda A_{i \rightarrow o} \cdot \lambda B_{i \rightarrow o} \cdot \lambda X_i \cdot \forall w_i \cdot (f X A W) \rightarrow (B W) \]
\[ \Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \quad = \lambda Q_{i \rightarrow o} \cdot \lambda A W_i \cdot \forall X_{i \rightarrow o} \cdot (Q X W) \]
\[ \Pi_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \quad = \lambda R_{i \rightarrow o} \cdot \lambda W_i \cdot \forall P_{i \rightarrow o} \cdot (R P W) \]

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

This mapping induces mappings \([\mathcal{F} V], [\mathcal{P} V]\) and \([\mathcal{F} \mathcal{I} M]\) of the sets \mathcal{F} V, \mathcal{P} V and \mathcal{F} \mathcal{I} M respectively.

Analyzing the validity of a translated formula \([\varphi]\) for a world represented by term \( t_i \) corresponds to evaluating the application \(([\varphi] t_i)\). In line with [4], we define \( \text{vld}_{(i \rightarrow o) \rightarrow o} = \lambda A_{i \rightarrow o} \cdot \forall S_i (A S) \).

With this definition, validity of a quantified conditional formula \( \varphi \) in CK corresponds to the validity of the corresponding formula \((\text{vld} [\varphi])\) 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 \( g = (g^\mathcal{V} : \mathcal{F} V \rightarrow D, g^\mathcal{P} V : \mathcal{P} V \rightarrow Q) \) be a variable assignment for a quantified conditional logic. We define the corresponding variable assignment \([g] = ([g^\mathcal{V}] : [\mathcal{F} V] \rightarrow D, [g^\mathcal{P} V] : [\mathcal{P} V] \rightarrow Q)\) for HOL so that \([g] (X_u) = [g] ([X]) = g(X)\) and \([g] (P_{i \rightarrow o}) = [g] ([P]) = g(P)\) for all \( X_u \in [\mathcal{F} V] \) and \( P_{i \rightarrow o} \in [\mathcal{P} V] \). Finally, a variable assignment \([g]\) is extended to an assignment for variables \( Z_\alpha \) of arbitrary type by choosing \([g] (Z_\alpha) = d \in D_\alpha\) arbitrary, if \( \alpha \neq u, i \rightarrow o \).

DEF. 5.2 (Henkin model \( \mathcal{H} \mathcal{M} \)). Given a quantified conditional logic model \( \mathcal{M} = \langle S, f, D, Q, I \rangle \). The Henkin model \( \mathcal{H} \mathcal{M} = \langle \{ D_\alpha \}_{\alpha \in \mathcal{F} I} \rangle \) for \( \mathcal{M} \) is defined as follows: \( D_i \) is chosen as the set of possible worlds \( S, D_\alpha \) is chosen as the first-order domain \( D \) (cf. definition of \([g^\mathcal{V}]\)), \( D_{i \rightarrow o} \) is chosen as the set of possible worlds \( Q \) (cf. definition of \([g^\mathcal{P} V]\))\footnote{To keep things simple, we identify sets with their characteristic functions.} and all other sets \( D_{\alpha \rightarrow \beta} \) are chosen as (not necessarily full) sets of functions from \( D_\alpha \) to \( D_\beta \). For all sets \( D_{\alpha \rightarrow \beta} \) the rule that everything denotes must be obeyed, in particular, we require that the sets \( D_{i \rightarrow o} \) and \( D_{i \rightarrow o} \rightarrow (i \rightarrow o) \) contain the elements \( 1_{k^\mathcal{V}}_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \) and \( 1_{f^\mathcal{V}}_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \) as characterized below.

The interpretation \( I \) is constructed as follows: \( i \) Let \( k_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = [k] \) for n-ary \( k \in \mathcal{F} \mathcal{I} M \) and let \( X^i_{\mathcal{V}} = [X^i] \) for \( X^i \in \mathcal{F} V, i = 1, \ldots , n \). We choose \( 1_{k_{(i \rightarrow o) \rightarrow (i \rightarrow o)}} \in D_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \) such that \( (1_{k_{(i \rightarrow o) \rightarrow (i \rightarrow o)}) ([g] (X^1), \ldots , [g] (X^n), w) = T \) for all worlds \( w \in D_1 \) such that \( \mathcal{M}, g, w \vDash k(X^1, \ldots , X^n) \), that is, if \( ([g^\mathcal{V}] (X^1), \ldots , [g^\mathcal{V}] (X^n)) \in I(k, w) \). Otherwise we choose \( (1_{k_{(i \rightarrow o) \rightarrow (i \rightarrow o)}) ([g] (X^1), \ldots , [g] (X^n), w) = F \).

(ii) We choose \( 1_{f^\mathcal{I}M}_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \in D_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \) such that \( (1_{f^\mathcal{I}M}_{(i \rightarrow o) \rightarrow (i \rightarrow o)}) (s, t, q) = T \) for all worlds \( s, t \in D_1 \) and \( q \in D_{i \rightarrow o} \) with \( t \in f(s, \{ x \in S \mid q(x) = T \}) \) in \( \mathcal{M} \). Otherwise we choose \( (1_{f^\mathcal{I}M}_{(i \rightarrow o) \rightarrow (i \rightarrow o)}) (s, t, q) = F \).

(iii) For all other constants \( s_\alpha \), choose \( 1_{s_\alpha} \) as arbitrary.\footnote{In fact, we may safely assume that there are no other typed constant symbols given, except for the symbol \( f^\mathcal{I}M_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \), the symbols \( k_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \), and the logical connectives.}

It is not hard to verify that \( \mathcal{H} \mathcal{M} \) is a Henkin model.
LEMMA 5.3. Let $\mathcal{H}^{\mathcal{M}}$ be a Henkin model for a selection function model $\mathcal{M}$. For all quantified conditional logic formulas $\delta$, variable assignments $g$ and worlds $s$ it holds: $\mathcal{M}, g, s \models \delta$ if and only if $\forall'([g][s]/S_i], ([\delta] S_i)) = T$

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

THEOREM 5.4 (Soundness and Completeness). $\models (vld [\phi])$ in HOL if and only if $\models \phi$ in CK

Proof. (Soundness) The proof is by contraposition. Assume $\not\models \phi$ in CK, that is, there is a model $\mathcal{M} = (S, f, D, Q, I)$, a variable assignment $g$ and a world $s \in S$, such that $\mathcal{M}, g, s \not\models \phi$. By Lemma 5.3 we have that $\forall'([g][s]/S_i], ([\phi] S_i)) = F$ in Henkin model $\mathcal{H}^{\mathcal{M}} = \langle \{D\alpha\} \alpha \in \mathcal{X}, I \rangle$ for $\mathcal{M}$. Thus, by Prop. 5.1 definition of vld and since $(\forall S_i. [\phi] S) = \beta\eta (vld [\phi])$ we know that $\forall'([g], (\forall S_i. [\phi] S)) = \forall'([g], (vld [\phi])) = F$. Hence, $\mathcal{H}^{\mathcal{M}} \not\models (vld [\phi])$, and thus $\not\models (vld [\phi])$ in HOL.

(Completeness) The proof is again by contraposition. Assume $\not\models (vld [\phi])$ in HOL, that is, there is a Henkin model $\mathcal{H} = \langle \{D\alpha\} \alpha \in \mathcal{X}, I \rangle$ and a variable assignment $\phi$ with $\forall'\phi, (vld [\phi]) = F$. 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 quantified conditional logic model $\mathcal{M}$ and that $\Phi = [g]$ for a corresponding quantified conditional logic variable assignment $g$. By Prop. 5.1 and since $(vld [\phi]) = \beta\eta (\forall S_i. [\phi] S)$ we have $\forall'([g], (\forall S_i. [\phi] S)) = F$, and hence, by definition of vld, $\forall'([g][s]/S_i], [\phi] S) = F$ for some $s \in D$. By Lemma 5.3 we thus know that $\mathcal{M}, g, s \not\models \phi$, and hence $\not\models \phi$ 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. Ruspini. 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, 2011. To appear. See also http://arxiv.org/abs/0905.2435.
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 syntax.

Satisfiability of this embedding is shown by the HOL reasoner Satallax\footnote{http://www.ps.uni-saarland.de/~cebrown/satallax/} in only 0.01 seconds.

\[
\begin{align*}
\text{thf}(f\_type,\text{type}, ( & f: \text{o} > ( \text{o} > \text{o} ) > ( \text{o} > \text{o} ) ) ) . \\
\text{thf}(\text{cnot}\_\text{type},\text{type}, ( & \text{cnot: ( } \text{o} > \text{o} ) > ( \text{o} > \text{o} ) ) ) . \\
\text{thf}(\text{cor}\_\text{type},\text{type}, ( & \text{cor: ( } \text{o} > \text{o} ) > ( \text{o} > \text{o} ) ) ) . \\
\text{thf}(\text{cttrue}\_\text{type},\text{type}, ( & \text{cttrue: } \text{o} > \text{o} ) ) .
\end{align*}
\]
Quantified Conditional Logics are Fragments of HOL

C. Benzmüller and V. Genovese

\[ \neg ( [X: i] : \text{true} ) \].

%---- 'false' in conditional logic
thf(cfcase_type,type,(cfcase: i > o)).

thf(cfcase_def,definition,(cfcase
= ( ^ [X: i] : false ) )).

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

thf(ccond_def,definition,(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 )).
Quantified Conditional Logics are Fragments of HOL  

\(\text{thf(cequiv_def, definition,}
\begin{align*}
( \text{cequiv} \\
= ( ^ [\Phi: \mu > \sigma, \Psi: \nu > \rho] : \\
( \text{cand} @ ( \text{cimpl} @ \Phi @ \Psi ) @ ( \text{cimpl} @ \Psi @ \Phi ) ))).
\end{align*}
\)

\(\%---- \text{ 'universal quantification (individuals)' in conditional logic}
\text{thf(cforall_ind_type,type,}
\begin{align*}
\text{cforall_ind:} & \ ( \mu > \nu > \rho ) > \nu > \rho \\
\text{cforall_ind, definition,}
\begin{align*}
& ( \text{cforall_ind} \\
= ( ^ [\Phi: \mu > \nu > \rho, W: \nu] : \\
! [X: \mu] : \\
( \Phi @ X @ W ) )).
\end{align*}
\end{align*}
\)

\(\%---- \text{ 'universal quantification (propositions)' in conditional logic}
\text{thf(cforall_prop_type,type,}
\begin{align*}
\text{cforall_prop:} & \ ( \nu > \rho ) > \nu > \rho \\
\text{cforall_prop, definition,}
\begin{align*}
& ( \text{cforall_prop} \\
= ( ^ [\Phi: \nu > \rho] : \\
! [P: \nu] : \\
( \Phi @ P @ W ))).
\end{align*}
\end{align*}
\)

\(\%---- \text{ 'existential quantification (individuals)' in conditional logic}
\text{thf(cexists_ind_type,type,}
\begin{align*}
\text{cexists_ind:} & \ ( \mu > \nu > \rho ) > \nu > \rho \\
\text{cexists_ind, definition,}
\begin{align*}
& ( \text{cexists_ind} \\
= ( ^ [\Phi: \mu > \nu > \rho] : \\
( \text{cnot} @ ( \text{cforall_ind} \\
@ ^ [X: \mu] : \\
( \text{cnot} @ ( \Phi @ X ) ) ) )).
\end{align*}
\end{align*}
\)

\(\%---- \text{ 'existential quantification (propositions)' in conditional logic}
\text{thf(cexists_prop_type,type,}
\begin{align*}
\text{cexists_prop:} & \ ( \nu > \rho ) > \nu > \rho \\
\text{cexists_prop, definition,}
\begin{align*}
& ( \text{cexists_prop} \\
= ( ^ [\Phi: \nu > \rho] : \\
( \text{cnot} @ ( \text{cforall_prop} \\
@ ^ [P: \nu] : \\
( \text{cnot} @ ( \Phi @ P ) ) ) )).
\end{align*}
\end{align*}
\)

\(\%---- \text{ 'validity' of a conditional logic formula}
\text{thf(valid_type,type,}
\begin{align*}
\text{valid:} & \ ( \nu > \rho ) > \rho \\
\text{valid_def, definition,}
\begin{align*}
& ( \text{valid} \\
= ( ^ [\Phi: \nu > \rho] : \\
( \text{cnot} @ ( \text{cforall_prop} \\
@ ^ [P: \nu] : \\
( \text{cnot} @ ( \Phi @ P ) ) ) )).
\end{align*}
\end{align*}
\)

\text{thf(valid_def, definition,}
\begin{align*}
& ( \text{valid} \\
= ( ^ [\Phi: \nu > \rho] : \\
( \text{cnot} @ ( \text{cforall_prop} \\
@ ^ [P: \nu] : \\
( \text{cnot} @ ( \Phi @ P ) ) ) )).
\end{align*}
\)}
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 our encoding assumes constant domain semantics.

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

\(^4\)http://www.leoprover.org
@ ( cforall_ind
  @ ^ [X: mu] :
    ( b @ X ))
@ ( cforall_ind
  @ ^ [X: mu] :
    ( ccond @ a @ ( b @ X )))

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