Embedding of Quantified Modal Logic in Higher Order Logic

Seminar Paper on Expressive Logics and their Automation

Alexander Steen,
a.steen@fu-berlin.de

Freie Universität Berlin
Department of Computer Science

Abstract. Church's Simple Theory of Types (STT, also referred to as classical higher-order logic) is an elegant and expressive formal language that can represent complex (higher-order) properties and formulae. In this paper, an encoding of second-order quantified modal logic (QML) in STT, due to C. Benzmüller and L. C. Paulsen, is discussed. Main results include the soundness and completeness of this encoding as well as its wide applicability and use for combining further logics. As an example, an practical encoding of Minimal Tense Logic K_t (bi-modal logic K with additional temporal axioms) in Isabelle/HOL is presented.

1 Introduction

In the course of the seminar, some "more expressive" alternatives to classical first-order logic have already been discussed. Whereas some logics are indeed more expressive in the formal sense, there are also logics that offer a more practical expressiveness, that is, the feature to state complex formulae compactly and more readable than classical FOL could express it. The downside, however, is revealed as soon as logics such as modal logics or conditional logics are to be automated using theorem provers (ATP), SMT solvers or interactive proof assistants: There exist only few ATP for this logics and for every experimental change in their semantics or further adjustments, they have to be adjusted or, in worst case, rewritten too.

In this seminar paper, an embedding technique is discussed that makes use of the simplicity and expressiveness of classical higher-order logic (STT) to encode quantified multimodal logic in STT. This idea allows to utilize ATPs for classical higher-order logic as "special purpose prover" for the embedded (and not necessarily higher-order) logic. As it turns out, several logics can be embedding using such an embedding technique; this paper follows closely the work of C. Benzmüller and L. Paulson [1].

The next two sections give a brief refresh on higher-order logic and quantified modal logic (and partly recaps previous seminar talks). Subsequently, the discussed encoding is sketched and a practical use case is given by embedding Minimal Tense Logic K_t in Isabelle/HOL.

2 Simple Theory of Types

In order to briefly recall the introductory seminar talk about higher-order logic, some aspects of STT such as its syntax and semantics are presented.

The Simple Type Theory, or *STT*, is a formal higher-order logic language developed by A. Church in 1970 which is grounded on the simply typed lambda calculus [2].

Every term of STT is associated a type. The set of base types $\mathfrak{T}$ is usually given by $\mathfrak{T} = \{o, \iota\}$, where o is chosen to be the type of truth values and ι the type of individuals. Throughout the remainder of this paper the set $\mathfrak{T} = \{o, \iota, \mu\}$ is considered, where μ is the designated type of possible worlds (STT meta-objects that correspond to worlds used in the QML semantics). The set of types T is then freely generated by the only type constructor $\rightarrow$, the function type constructor.
Note that the function type constructor is right-associative: For $\alpha_1, \ldots, \alpha_n \in T$, $\alpha_1 \rightarrow \alpha_2 \rightarrow \cdots \rightarrow \alpha_n$ is rather written than $\alpha_1 \rightarrow (\alpha_2 \rightarrow (\cdots \rightarrow \alpha_n) \ldots)$.

Example 1 (Some types).

(i) $\iota \rightarrow \iota \rightarrow o$ — Type of a binary relation over individuals

(ii) $o \rightarrow o$ — Type of an unary logical connective

(iii) $(\iota \rightarrow o) \rightarrow o$ — Type of some higher-order function

┘

The type of a term is written as index, e.g. X_α for X that has type $\alpha \in T$. A type index may be omitted if it is clear from the context (e.g. from binder).
The syntax of the STT language L is given by

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

where application of terms is left-associative and lambda-abstraction is right-associative and $\alpha, \beta \in T$. Top-level-brackets are omitted. p_α are constants of type α whereas X_α denote variables.

Note that quantification is expressed by the $\Pi_{(\alpha \rightarrow o) \rightarrow o}$ connective: Since STT offers a general binding mechanism via lambda abstraction, universal quantification and existential quantification symbols can be regarded as defined terms and are then given by $\forall X_\alpha . s_o := \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda X_\alpha . s_o)$ and $\exists X_\alpha . s_o := \neg \forall X_\alpha . \neg s_o$ respectively.

The semantics of higher-order logic was already thoroughly discussed in another seminar talk and can also be found in the literature (e.g. [3]). For any interpretation $(\{D_\alpha\}_{\alpha \in T}, I)$ the set of denotations for Boolean values is, here, given by $D_o = \{\top, \bot\}$ (representing truth and falsehood respectively). The function I is chosen such that the logical connectives $\neg_{o\to o}$, $\vee_{o\to o\to o}$, $=_{\alpha\to\alpha\to o}$ and $\Pi_{(\alpha\to o)\to o}$ have their usual meaning [1].

A formula $A \in L$ is called Henkin-valid, written $\models^{STT} A$, if A is *valid* in every Henkin model. Validity is defined as usual; see previous seminar talk about higher-order logics for details.

3 Quantified Modal Logic

As a further prerequisite, main aspects of quantified (second-order) multimodal logic, QML, are recaptured in the following. In contrast to the previous section, the semantics are here described in more detail because they will intensively be handled with in Sect. 4. The language description is taken from [1] and inspired by [4].

For the syntax of the QML language, some sets need no be fixed first: Let I be some index set, containing the identifier of the different accessibility relations. And since QML is a second-order language, we need to distinguish the two sorts of variables that can be quantified over: Let IV and PV be sets of individual variables and propositional variables respectively and let furthermore be SYM a set of predicate symbols (of any finite arity).
The syntax of QML languages are then given by

$$s, t ::= P \mid f(X^1, \ldots, X^n) \mid \neg s \mid s \vee t \mid \Box_r s \mid \forall X.s \mid \forall P.s$$

where $r \in I, P \in PV, k \in \mathrm{SYM}, X, X^i \in IV$.

The idea and origins of Kripke's semantics of possible worlds [5] were also intensively discussed before, as well as the development of different modes for quantification (constant vs. varying domain) and further distinguishing features. In the embedding, only constant-domain quantification is considered [1]. As of section 4, the so-called $QK\pi$ models of QML will be related to Henkin models of the STT language. Hence, they are introduced more thoroughly here:
The quantified (second-order) model-equivalent for a (multi-) modal logic K is defined in two steps. The weaker structure QKπ^- is defined first along with assignments and validity, the actual QKπ models are then formulated as specializations.

Definition 1 (QKπ^- model). *Let W be a set of worlds and $\{R_r\}_{r\in I}$ a collection of accessibility relations among them (i.e. such that $(W, \{R_r\}_{r\in I})$ is a multimodal frame). Let further be D a non-empty set, P a non-empty collection of subsets of W and I_w, for each world $w \in W$, a functions that assigns to every n-ary relation symbol $k \in SYM$ a n-ary relation over D in w.*
Then, a $QK\pi^-$ model is a quintuple $(W, \{R_r\}_{r\in I}, D, P, \{I_w\}_{w\in W})$. ⌟

The accessibility relations R_r and the interpretation functions I_r are indexed by a relation identifier since we are dealing with a multimodal logic; each $r \in I$ can be seen as independent agent or actor. The set D (the first-order domain) contains all the elements that individual variables may carry (as value), the set

P (the propositional domain) fixes the subsets of worlds that can be used as carrier for propositions (i.e. the worlds in which a certain proposition is true).

Definition 2 (Variable assignment). *A variable assignment g is a pair $g = \left(g^{iv}, g^{pv}\right)$ where $g^{iv} : IV \to D$ and $g^{pv} : PV \to P$ are mappings for the corresponding variables to their domains.*

For a model M, a variable assignment g and a world w, validity of a formula s, written $M, g, w \models s$, as well as all accompanying terms of validity, are defined as usual.

The now following specialization of QKπ^- models is used as semantic structure for QML languages throughout the paper.

Definition 3 (QKπ model). *A QKπ^- model $M = (W, \{R_r\}_{r\in I}, D, P, \{I_w\}_{w\in W})$ is a QKπ model if for every formula $s \in$ QML and every variable assignment g, the extension of s is contained in P, that is, $\{w \in W | M, g, w \models s\} \in P$.*

A formula s is called QKπ-valid, denoted $\models^{\mathrm{QK}\pi} s$, if it is valid in all QKπ models.

4 Embedding of QML in STT

The idea. Boolean formulae are represented by the type o. Their modal equivalent in STT are also dependent on the world the formula is evaluated in, hence yielding the type $\mu \to o$. In other words, the idea is that modal formulae are considered functions that take the current world as parameter and return the truth value of the formula in that world.

In order to find a mapping from QML formulae to STT terms, first the types of the corresponding operators are identified in table 1. It is easy to see that any place logically containing a Boolean parameter, now has to be of type $\mu \to o$. Since the accessibility relation identifier are now in scope of the language, only one box operator needs to be modeled. The accessibility relation r of type $\mu \to \mu \to o$ can then just be applied as function argument, yielding the corresponding box operator $\Box_r$. Much alike, the quantifiers can be modelled as constant symbols, where $\forall^\iota$ and $\forall^{\mu\to o}$ represent the quantification over individual variables and propositional variables respectively.

QML term	Type signature in STT
P	$\mu \to o$
f	$\iota^n \to \mu \to o$
$\neg$	$(\mu \to o) \to \mu \to o$
$\vee$	$(\mu \to o) \to (\mu \to o) \to \mu \to o$
$\Box$	$(\mu \to \mu \to o) \to (\mu \to o) \to \mu \to o$
Π^ι	$(\iota \to \mu \to o) \to \mu \to o$
$\Pi^{\mu\to o}$	$((\mu \to o) \to \mu \to o) \to \mu \to o$

Table 1: Types of the QML terms

Table 2 contains the actual definitions of the previously mentioned operators. These operators are then used as part of the *embedded QML language*, called QMLSTT.

QML term	Term in STT
P	$P_{\mu\to o}$
f	$f_{\iota^n\to\mu\to o}$
$\neg$	$\lambda\phi_{\mu\to o}.\lambda W_\mu.\phi W$
$\vee$	$\lambda\phi_{\mu\to o}.\lambda\psi_{\mu\to o}.\lambda W_\mu.\phi W \vee \psi W$
$\Box$	$\lambda R_{\mu\to\mu\to o}.\lambda\phi_{\mu\to o}.\lambda W_\mu.\forall V_\mu.\neg(RWV) \vee \phi V$
Π^ι	$\lambda\phi_{\iota\to\mu\to o}.\lambda W_\mu.\forall X_\iota.\phi XW$
$\Pi^{\mu\to o}$	$\lambda\phi_{(\mu\to o)\to\mu\to o}.\lambda W_\mu.\forall P_{\mu\to o}.\phi PW$

Table 2: Translation of the QML terms

The definitions of the operators are straight-forward: QMLSTT's $\neg$ and $\vee$ take one and two terms as their first parameters, just as the standard definitions do. Additionally, they take (just as any other QMLSTT operator) the current world as last parameter. Together with the information of the current world, the truth value (of type o) can then be calculated by standard application. The other operators are defined similarly.
For the QMLSTT language, we fix sets IVSTT, PVSTT, ISTT, SYMSTT with members of type ι, $\mu\to o$, $\mu\to\mu\to o$ and $\iota^n\to\mu\to o$ respectively (analogously to the sets of QML). The set of QMLSTT propositions (of type $\mu\to o$) is then defined inductively as follows:

(i) Each $P_{\mu\to o}\in$ PVSTT is a QMLSTT proposition
(ii) For any n-ary symbol $f_{\iota^n\to\mu\to o}\in$ SYMSTT and $X^1_\iota,\dots,X^n_\iota\in$ IVSTT, the term $(fX^1,\dots,X^n)_{\mu\to o}$ is a QMLSTT proposition
(iii) For any QMLSTT propositions ϕ and ψ, $\phi\vee\psi$ and $\neg\phi$ are also QMLSTT propositions
(iv) For each $r_{\mu\to\mu\to o}\in$ ISTT and every QMLSTT proposition ϕ, the term $\Box\, r\,\phi$ is also a QMLSTT proposition
(v) For any $X_\iota\in$ IVSTT, $P_{\mu\to o}\in$ PVSTT and any QMLSTT proposition ϕ, the terms $\Pi^\iota(\lambda X_\iota.\phi_o)$ and $\Pi^{\mu\to o}(\lambda P_{\mu\to o}.\phi_o)$ are also QMLSTT propositions

Analogously to Sect. 2, $\forall X_\iota.\phi$ and $\forall P_{\mu\to o}.\phi$ may be written instead of $\Pi^\iota(\lambda X_\iota.\phi_o)$ and $\Pi^{\mu\to o}(\lambda P_{\mu\to o}.\phi_o)$, respectively.
Since we can partially apply arguments to functions, the term $\Box_r$ can, as usual, be used as a shorthand for $\Box r \equiv \lambda\phi_{\mu\to o}.\lambda W_\mu.\forall V_\mu.\neg(rWV)\vee\phi V$ (i.e. the box already applied to its first argument r).

Reflecting the validity of QML, QMLSTT validity is defined as follows: A proposition $\phi_{\mu\to o}$ is called valid if for all possible worlds w_μ it holds that

$\phi_{\mu\to o} w_\mu = \top$. The abbreviation *valid* is defined in order to give a useful shortcut for validity tasks:

Definition 4 (QMLSTT Validity). *The STT function valid is given by* $valid = \lambda\phi_{\mu\to o}.\forall W_\mu.\phi W$ ⌟

The *valid* function can be used to compactly express validity tasks of QMLSTT propositions, i.e. for grounding the QMLSTT propositions to STT.

Mapping of the languages and models. The mapping of QML languages to QMLSTT languages is straight-forward by choosing the sets IVSTT, PVSTT, ISTT, SYMSTT accordingly to the sets IV, PV, I and SYM (respectively). The mapping function is denoted $\dot{\_}$, i.e. if $r \in I$, then $\dot{r} \in$ ISTT. The according backward mapping function (needed for the completeness result) is denoted $\overline{\_}$.

Also, a $QK\pi$ model can be lifted to a STT Henkin model by choosing the domains D_ι, D_μ, $D_{\mu\to o}$, $D_{\mu\to\mu\to o}$ and all $D_{\alpha\to\beta}$ as the set D of individuals, the set W of possible worlds, the set P of sets of possible worlds, the set R of accessibility relations and the set of functions from D_α to D_β respectively The interpretation function I is chosen such that the predicate symbols and accessibility relations of the QML language are accordingly lifted to the QMLSTT language. Since this is rather technical, the complete mapping construction is omitted and can be found in [1].

Soundness and completeness results. Only the results themselves are stated here, the proofs can again be found in [1].

Theorem 1 (Soundness for $QK\pi$ Semantics). *Let s be a QML proposition and let $s_{\mu\to o} = \dot{s}$ be the corresponding QMLSTT proposition.*
If $\models^{STT}$ (valid $s_{\mu\to o}$) then $\models^{QK\pi}$ s. ⌟

Theorem 2 (Completeness for $QK\pi$ models). *Let $s_{\mu\to o}$ be a QMLSTT proposition and let $s = \overline{s}_{\mu\to o}$ be the corresponding QML proposition.*
If $\models^{QK\pi}$ s then $\models^{STT}$ (valid $s_{\mu\to o}$). ⌟

5 Practical Embedding using Isabelle/HOL

In this section, the results of Sect. 4 are exploited to perform an actual practical embedding of a quantified version of Minimal Tense Logic K_t in Isabelle/HOL [6].

Tense Logic [7] is a bi-modal logic (i.e. with two independent modal operators) that employs four modalities for reasoning over time. The *strong tense operators* H and G and the *weak tense operators* P and F have the following meaning:

$H\phi$:	ϕ held at all times throughout the past
$G\phi$:	ϕ holds at all times throughout the future
$P\phi$:	ϕ held at some time in the past
$F\phi$:	ϕ holds at some time in the future

The weak tense operators are dual to the strong operators in the sense that are mutually definable: It holds that $\neg H \neg \phi \equiv P\phi$ and $\neg G \neg \phi \equiv F\phi$. Tense Logic has been introduced and intensively researched by A. Prior (e.g. [8], [9]). There exist several extensions, yet in this section only the minimal tense logic K_t is considered. Logic K_t is obtained by adding the two axioms schemes 1 and 2 to a standard bi-modal logic K (i.e. two modal operators each with modal context K).

$$\begin{aligned} 1: \quad & \phi \implies HF\phi \\ 2: \quad & \phi \implies GP\phi \end{aligned}$$

In the following, an excerpt of the embedding of K_t in Isabelle/HOL is presented. It is primarily based on the original encoding of QML in Isabelle/HOL [10], but changes the properties of the accessibility relation to match logic K_t. To intuition of possible worlds is here adjusted to describe worlds as points in time (of type τ) and the accessibility relation $<_t$ as (strict) partial ordering of these time points (c.f. Listing 1.1).

```
typedecl i    -- "the type for individuals"
typedecl τ    -- "the type for times (worlds)"
consts order :: "τ ⇒ τ ⇒ bool" (infixr "t<" 70)

axiomatization where
  trans: "∀ x.∀ y. ∀z. x t< y ∧ y t< z ⟶ x t< z" and
  asym: "∀x. ∀y. (x t< y) ⟶ ¬(y t< x)"

type_synonym σ = "(τ ⇒ bool)"
```

Listing 1.1: The types and ordering relation of K_t

The type synonym σ is adopted from [10] for shortening the signatures of Listing 1.2. The definition of validity (c.f. Definition 4) is omitted, its syntactical representation in Isabelle in given by the brackets [and], which enclose the actual argument. So, *valid* ϕ is written `[ϕ]`. The axiomatization 1 and 2 (see further above) is here implied by the properties of the underlying ordering relation $<_t$ (identified by `order` in Isabelle/HOL) instead of being explicitly denoted in terms of tense-logical formulae. This is due to the fact that a strict ordering relation was chosen and there is no formula that corresponds to irreflexibility [11].

```
abbreviation tnot :: "σ ⇒ σ" ("t¬")
    where "t¬ ϕ ≡ (λt. ¬ ϕ t)"
abbreviation tand :: "σ ⇒ σ ⇒ σ" (infixr "t∧" 51)
    where "ϕ t∧ ψ ≡ (λt. ϕ t ∧ ψ t)"
abbreviation tor :: "σ ⇒ σ ⇒ σ" (infixr "t∨" 50)
    where "ϕ t∨ ψ ≡ (λt. ϕ t ∨ ψ t)"
abbreviation timplies :: "σ ⇒ σ ⇒ σ" (infixr "t⇒" 49)
    where "ϕ t⇒ ψ ≡ (λt. ϕ t ⟶ ψ t)"
abbreviation tforall :: "('a ⇒ σ) ⇒ σ" ("∀")
    where "∀ ϕ ≡ (λt. ∀x. ϕ x t)"
```

```
abbreviation texists :: "('a ⇒ σ) ⇒ σ" ("∃")
    where "∃ ϕ ≡ (λt. ∃x. ϕ x t)"
...
abbreviation tP :: "σ ⇒ σ" ("P")
    where "P ϕ ≡ (λt. ∃t'. t' t< t ∧ ϕ t')"
abbreviation tF :: "σ ⇒ σ" ("F")
    where "F ϕ ≡ (λt. ∃t'. t t< t' ∧ ϕ t')"
abbreviation tH :: "σ ⇒ σ" ("H")
    where "H ϕ ≡ (λt. ∀t'. t t< t' ∨ ϕ t')"
abbreviation tG :: "σ ⇒ σ" ("G")
    where "G ϕ ≡ (λt. ∀t'. t' t< t ∨ ϕ t')"
```

Listing 1.2: An excerpt of the logical connective definitions of K_t

The operators are defined in the previously discussed manner (c.f. Sect. 4), the definitions of P, F, H and G reflect their intended meaning: $H\phi_{\tau\to o}$ holds if and only if $\phi_{\tau\to o}t_\tau$ holds for every past time t_τ. Analogously for the remaining operators.
As depicted in Listing 1.3, Metis can now be used to check that the axioms 1 and 2 indeed hold:

```
theorem A1: "[∀(λϕ. ϕ t→ (H (F ϕ)))]" by metis
theorem A2: "[∀(λϕ. ϕ t→ (G (P ϕ)))]" by metis
```

Listing 1.3: Metis can verify the axioms 1 and 2

Now that the embedding is complete and the axioms are set, further proof tasks can be automated using the classical higher-order logic proof assistant Isabelle/HOL. As an example, certain correspondences can be verified automatically, such as

$$\text{leftserial } <_t \iff H\Phi \Rightarrow P\Phi$$

where leftserial is a predicate defined by

$$\text{leftserial} = \lambda order_{\tau\to\tau\to o}.\forall t_\tau.\exists t'_\tau.t' <_t t$$

The automated proof is split in two lemma and formulated as shown in Listing 1.4

```
lemma L1a:
   assumes "[∀(λΦ.(H Φ) t→ (P Φ))]"
   shows "∀t.∃t'. t' t< t"
  by (metis asym assms A2 trans)


lemma L1b:
   assumes "∀t.∃t'. t' t< t"
   shows "[∀(λΦ.(H Φ) t→ (P Φ))]"
  by (metis assms asym)
```

Listing 1.4: Lemma for correspondence leftserial $<_t \Leftrightarrow (H\Phi \Rightarrow P\Phi)$

Listing 1.5 shows a proof task that Metis is correctly refuting: It cannot be shown from the axioms that the ordering on type τ is dense. Nitpick can even give a counter-model to verify this result.

```
lemma 3:
   shows "∀t.∀t'. (t t< t' ⟶ (∃t''. t t< t'' ∧ t'' t< t'))"
  by metis
  oops
```

Listing 1.5: Isabelle/HOL (correctly) cannot proof the density of τ

As it can be seen, proof tasks in modal logic (here Minimal Tense Logic K_t) can be automated using a classical higher-order prover.

6 Conclusion and Related work

In this seminar paper, the embedding of (quantified) modal logic in higher-order logic was sketched. It can be shown that the embedding of so-called $QK\pi$ models of quantified modal logic is sound and complete with respect to Henkin models of STT. Since the construction of the discussed embedding is relatively straightforward, it can be adopted for logics with Kripke-like semantics and possibly even another ones. As an example, there are approaches to embed Conditional Logic [12] and Intuitionistic Logics [13]. As a further step, logics can even be combined in the embedded setting [14] to achieve e.g. multiple modalities within different modal contexts or combine temporal logic with epistemic modalities.
As a major benefit of the embedding approach, reasoning within (and even about) embedded logics can be done with classical higher-order ATPs.

References

1. Benzmüller, C., Paulson, L.: Quantified multimodal logics in simple type theory. Logica Universalis (Special Issue on Multimodal Logics) **7**(1) (2013) 7–20
2. Church, A.: A formulation of the simple theory of types. Journal of Symbolic Logic **5**(2) (June 1940) 56–68
3. Andrews, P.: Church's type theory. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Spring 2014 edn. (2014)
4. Fitting, M.: Interpolation for first order $s5$. Journal of Symbolic Logic **67**(2) (06 2002) 621–634
5. Garson, J.: Modal logic. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Spring 2013 edn. (2013)
6. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Volume 2283 of LNCS. Springer (2002)
7. Galton, A.: Temporal logic. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Fall 2008 edn. (2008)
8. Prior, A.: Time and Modality. Oxford University Press (2003)
9. Prior, A.: Past, Present and Future. Oxford books. OUP Oxford (1967)

10. Benzmüller, C., Paleo, B.W.: Gödel's god in isabelle/hol. Archive of Formal Proofs (November 2013) `http://afp.sf.net/entries/GoedelGod.shtml`, Formal proof development.
11. van Benthem, J.: Papers on time and tense. Kluwer Academic Publishers, Dordrecht, Boston and London (1983)
12. Benzmüller, C., Gabbay, D., Genovese, V., Rispoli, D.: Embedding and automating conditional logics in classical higher-order logic. Annals of Mathematics and Artificial Intelligence **66**(1-4) (2012) 257–271
13. Benzmüller, C., Paulson, L.: Multimodal and intuitionistic logics in simple type theory. The Logic Journal of the IGPL **18**(6) (2010) 881–892
14. Benzmüller, C.: Simple type theory as framework for combining logics. In: Contest paper at the World Congress and School on Universal Logic III (UNILOG), Lisbon, Portugal (2010) The conference had no published proceedings; the paper is available as arXiv:1004.5500v1.