Noname manuscript No.
(will be inserted by the editor)

Cut-Elimination for Quantified Conditional Logic

Christoph Benzmüller

Received: Dec 18, 2014 / Accepted: Feb 4, 2016

Abstract A semantic embedding of quantified conditional logic in classical higher-order logic is utilized for reducing cut-elimination in the former logic to existing results for the latter logic.

The presented embedding approach is adaptable to a wide range of other logics, for many of which cut-elimination is still open. However, special attention has to be payed to cut-simulation, which may render cut-elimination as a pointless criterion.

Keywords cut-elimination; quantified conditional logics; classical higher-order logic; semantic embedding; cut-simulation

1 Introduction

The development of cut-free calculi for expressive logics, for example, quantified non-classical logics, is usually a non-trivial task. However, for a wide range of logics there exists a surprisingly elegant and uniform solution: By modeling and studying these logics as fragments of classical higher-order logic (HOL) [1,14] — a research direction that has recently been proposed [12] — existing cut-elimination results for HOL can often be reused. In this article the embedding approach is exemplarily utilized for proving cut-elimination for quantified conditional logics (QCL).

Conditional logics [61,30], known also as logics of normality or typicality, have many applications including counterfactual reasoning, default reason-

This work has been supported by the German National Research Foundation (DFG) under grant BE 2501/9-1,2.

Christoph Benzmüller
Freie Universität Berlin, Dep. of Mathematics and Computer Science, 14095 Berlin, Germany
Stanford University, CSLI/Cordura Hall, Stanford, CA 94305-4115, U.S.
E-mail: c.benzmueller@fu-berlin.de

ing, deontic reasoning, metaphysical modeling, action planning, and reasoning about knowledge. While there is broad literature on propositional conditional logics only a few authors have addressed first-order extensions, those include Delgrande [37,36] and Friedman et al. [40].

QCL as addressed in this article includes constant- and variable-domain first-order quantification and combines it with quantification over propositional variables. Such a rich combination has not been adressed in the literature. In particular, cut-elimination for this logic, or similarly expressive QCLs, has not been studied yet. Only for propositional conditional logics some related work is available, for example, by Pattinson and Schröder [54] and by Rasga [57]; cf. also the references therein.

Delgrande [37,36] motivates the development and use of first-order conditional logics and he points to the problems of naive, constant domain quantification in this context. His framework, therefore, supports variable and constant domain quantification over individuals, but it does not consider propositional quantification. Moreover, cut-elimination is not addressed in his work.

Recently, a selection of Delgrande's motivating examples has been automated with higher-order automated theorem provers [10], and the embedding approach has provided the theoretical foundation for this practical work. Here we utilize the embedding approach further for showing cut-elimination.

First-order quantification in conditional logics has been studied also by Friedman et al. [40]. Their focus is on default reasoning and they develop a subjective and statistical first-order logic of conditionals. Their conditional logic semantics is based on plausibility measures. Again, cut-elimination is not addressed.

Previous work [17] has studied the embedding of propositional conditional logics in HOL. This work was subsequently extended to include first-order and propositional quantifiers [10,18]. However, these papers primarily focus on demonstrating the practical feasibility of the embedding approach. Most importantly, cut-elimination and cut-simulation is addressed in none of them.

The work presented here summarises and extends two unpublished workshop papers [11,18] and parts of a non-refereed, invited paper [13].

The remainder of the article is organized as follows: §2 introduces QCL. HOL, cut-elimination for HOL and cut-simulation are addressed in §3. §4 then shows how cut-elimination for QCL can be reduced to cut-elimination for HOL; however, in some cases cut-simulation effects apply. §5 outlines how the embedding approach can be utilized in practice. The article is concluded in §6.

2 Quantified Conditional Logic

Propositional conditional logic is extended here with quantification over *propositional variables* and over *individual variables*. Regarding the latter *constant domains* (every possible world has the same domain) and *varying domains* (different possible worlds may have different domains) are supported; in this regard the framework below is related to that of Delgrande. However, the inclusion of propositional quantification is novel. The gained expressivity is of crucial significance for the automation of QCLs with HOL provers and model finders [10].

To define the formulas of QCL we start out with a denumerable set of first-order (individual) variables IV, a denumerable set of propositional variables PV, and a denumerable set of predicate symbols SYM (of any arity).

The *atomic formulas of QCL* are made up from propositional variables $P \in \mathrm{PV}$ and from propositions $k(X^1,\ldots,X^n)$, where $k \in \mathrm{SYM}$ is an n-ary $(n \geq 0)$ predicate symbol which is applied to individual variables $X^1,\ldots,X^n$ (with $X^i \in \mathrm{IV}$ for $0 \leq i \leq n$).

The set of *QCL formulas* is given as the smallest set of formulas obeying the following conditions. Every atomic formula of QCL is also a QCL formula. Moreover, given any QCL formulas φ and ψ, then $\neg\varphi$ (negation), $\varphi \vee \psi$ (disjunction), $\varphi \Rightarrow \psi$ (conditionality), $\forall^{co} X\varphi$ (constant domain quantification), $\forall^{va} X\varphi$ (varying domain quantification) and $\forall P\varphi$ (propositional quantification) are also QCL formulas.

From the selected set of primitive connectives above, other logical connectives can be introduced as abbreviations: for example, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ (material implication), $\varphi \longleftrightarrow \psi$ and $\Box\varphi$ abbreviate $\neg(\neg\varphi \vee \neg\psi)$, $\neg\varphi \vee \psi$, $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ and $\neg\varphi \Rightarrow \varphi$, respectively. $\forall^{co}$ and $\forall^{va}$ are associated with constant domain and variable domain quantification. For $* \in \{co, va\}$, $\exists^* X\varphi$ abbreviates $\neg\forall^* X\neg\varphi$. Syntactically, QCL 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, different formalizations have been proposed [51]. Here we build on *selection function semantics* [61,30], which is based on possible world structures and has been successfully used in [52] to develop proof methods for some propositional CLs.

An *interpretation* is a structure $M = \langle S, f, D, D', Q, I\rangle$ where, S is a set of items called possible worlds, $f : S \times 2^S \mapsto 2^S$ is the selection function, D is a non-empty set of *individuals* (the constant first-order domain), D' is a function that assigns a non-empty subset $D'(w)$ of D to each possible world w (the $D'(w)$ are the varying domains), Q is a non-empty collection of subsets of S (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* $g = (g^i, g^p)$ is a pair of maps where, $g^i : \mathrm{IV} \mapsto D$ maps each individual variable in IV to an object in D, and $g^p : \mathrm{PV} \mapsto 2^D$ maps each propositional variable in PV to a set of worlds in Q.

Satisfiability of a formula φ for an interpretation $M = \langle S, f, D, D', Q, I\rangle$, a world $s \in S$, and a variable assignment $g = (g^i, g^p)$ is denoted by $M, g, s \models \varphi$ 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 $\langle g^i(X^1), \ldots, g^i(X^n)\rangle \in I(k, s)$

$M, g, s \models P$ if and only if $s \in g^p(P)$

$M, g, s \models \neg\varphi$ 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 $M, g, s \models \psi$

$M, g, s \models \forall^{co} X\varphi$ if and only if $M, ([d/X]g^i, g^p), s \models \varphi$ for all $d \in D$

$M, g, s \models \forall^{va} X\varphi$ if and only if $M, ([d/X]g^i, g^p), s \models \varphi$ for all $d \in D'(s)$

$M, g, s \models \forall P\varphi$ if and only if $M, (g^i, [p/P]g^p), s \models \varphi$ 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 $M = \langle S, f, D, D', Q, I\rangle$ is a QCL *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. This requirement, which is inspired by Fitting [38], Def. 3.5, ensures a natural correspondence to Henkin models in HOL.

As usual, a QCL formula φ is *valid in a QCL model* $M = \langle S, f, D, D', Q, I\rangle$, denoted with $M \models^{QCL} \varphi$, if and only if for all worlds $s \in S$ and variable assignments g holds $M, g, s \models \varphi$. A formula φ is *valid*, denoted $\models^{QCL} \varphi$, if and only if it is valid in every QCL model.

Most interestingly, QCL subsumes normal (quantified) modal logics, since $\Box\varphi$ can be defined as an abbreviation for $\neg\varphi \Rightarrow \varphi$; cf. [61].

Note that f is defined to take $[\varphi]$, called the *proof set* of φ with respect to a given QCL model M, instead of φ. This approach has the consequence of forcing the so-called *normality* property: given a QCL model M, if φ and φ' are equivalent, i.e., they are satisfied in the same set of worlds, then they index the same formulas with respect to the $\Rightarrow$ modality.

The axiomatic counterpart of the normality condition is given by the rule RCEA, which expresses a replacement property for equivalent formulas on the left-hand side of a conditional formula:

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

Moreover, it can be easily shown that the above semantics forces also the following rules to hold (RCEC expresses a right-hand side replacement property analogous to RCEA, and RCK expresses compatibility of the right-hand side of conditional formulas with conjunction):

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

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

We refer to QCK (cf. CK in [31]) as the minimal QCL closed under rules RCEA, RCEC and RCK. In what follows, only QCLs extending QCK are considered.

The admissibility of the rules RCEA, RCK and RCEC for QCL can be quickly proved by automated theorem provers when utilizing the embedding approach. That is, these rules are automatically entailed in the approach and can therefore be omitted; cf. [17] and §5.

3 Classical Higher-order Logic

Predicate logic with higher-order quantification was developed first by Frege in his Begriffsschrift [39] and by Russell in his ramified theory of types [58], which was later simplified by others, including Chwistek and Ramsey [56, 35], Carnap, and finally Church [34] in his simple theory of types, also referred to as classical higher-order logic (HOL).

HOL bases both terms and formulas on simply typed λ-terms and the equality of terms and formulas is given by equality of such λ-terms. The use of the λ-calculus has some major advantages. For example, λ-abstractions over formulas allow the explicit naming of sets and predicates, something that is achieved in set theory via the comprehension axioms. Another advantage is, that the complex rules for quantifier instantiation at first-order and higher-order types is completely explained via the rules of λ-conversion (the so-called rules of α-, β-, and η-conversion) which were proposed earlier by Church [32, 33]. These two advantages are heavily exploited in our embedding of QCL in HOL in §4.

For defining the language HOL, we first introduce the set T of *simple types*: As usual, we assume that T is freely generated from a set of *basic types* $BT \supseteq \{o, i\}$ using the function type constructor $\rightarrow$. o denotes the (bivalent) set of Booleans, and i a non-empty set of individuals. Further base types may be added, and we will in fact exploit a third base type u in §4.

For the definition of HOL, we start out with a family of denumerable sets of typed constant symbols $(C_\alpha)_{\alpha \in T}$, called *signature*, and a family of denumerable sets of typed variable symbols $(V_\alpha)_{\alpha \in T}$. We employ Church-style typing, where each term t_α explicitly encodes its type information in subscript α.

The language of HOL is given as the smallest set of terms obeying the following conditions. Every typed constant symbol $c_\alpha \in C_\alpha$ and every typed variable symbol $X_\alpha \in V_\alpha$ are HOL terms of type α. If $X_\alpha \in V_\alpha$ is a typed variable symbol and s_β is an HOL term of type β, then $(\lambda X_\alpha s_\beta)_{\alpha\rightarrow\beta}$, called *abstraction*, is an HOL term of type $\alpha \rightarrow \beta$. If $s_{\alpha\rightarrow\beta}$ and t_α are HOL terms of types $\alpha \rightarrow \beta$ and α, respectively, then $(s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$, called *application*, is an HOL term of type β.

The above definition encompasses the simply typed λ-calculus. In order to extend this base framework into HOL we simply ensure that the signature $(C_\alpha)_{\alpha \in T}$ provides a sufficient selection of primitive logical connectives. Without loss of generality, we here assume the following *primitive logical connectives* to be part of the signature: $\neg_{o\rightarrow o} \in C_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o} \in C_{o\rightarrow o\rightarrow o}$ and $\Pi_{(\alpha\rightarrow o)\rightarrow o} \in C_{(\alpha\rightarrow o)\rightarrow o}$ (for each type α). The denotation of these special constant symbols is fixed below according to their intended meaning. HOL is thus a logic of terms in the sense that the *formulas of HOL* are given as the terms of type o.

In addition to the primitive logical connectives selected above, we could assume *choice operators* $\epsilon_{(\alpha\to o)\to\alpha} \in C_{(\alpha\to o)\to\alpha}$ (for each type α) and *primitive equality* $=_{\alpha\to\alpha\to\alpha} \in C_{\alpha\to\alpha\to\alpha}$ (for each type α), abbreviated as $=^\alpha$, in the signature. We are not pursuing this here.

Type information as well as brackets may be omitted if obvious from the context. For example, we may write $(s \vee t)$ instead of $((\vee_{o\to o\to o} s_o) t_o)$.

From the selected set of primitive connectives, other logical connectives can be introduced as abbreviations: for example, $\varphi \wedge \psi$, $\varphi \to \psi$, and $\varphi \longleftrightarrow \psi$ abbreviate $\neg(\neg\varphi \vee \neg\psi)$, $\neg\varphi \vee \psi$, and $(\varphi \to \psi) \wedge (\psi \to \varphi)$, respectively.

Binder notation $\forall X_\alpha\, s_o$ is used as an abbreviation for $\Pi_{(\alpha\to o)\to o} \lambda X_\alpha s_o$.

Equality can actually be defined in HOL by exploiting Leibniz' principle, expressing that two objects are equal if they share the same properties. *Leibniz equality* $\doteq^\alpha$ at type α is thus defined as $s_\alpha \doteq^\alpha t_\alpha := \forall P_{\alpha\to o}(\neg P s \vee P t)$.

Each occurrence of a variable in a term is either bound by a λ or free. We use $free(s)$ to denote the set of free variables of s (i.e., variables with a free occurrence in s). We consider two terms to be *equal* if the terms are the same up to the names of bound variables (i.e., we consider α-conversion implicitly). A term s is *closed* if $free(s)$ is empty.

Substitution of a term s_α for a variable X_α in a term t_β is denoted by $[s/X]t$. Since we consider α-conversion implicitly, we assume the bound variables of t avoid variable capture.

Well-known operations and relations on HOL terms include *$\beta\eta$-normalization* and *$\beta\eta$-equality*, denoted by $s =_{\beta\eta} t$, *β-reduction* and *η-reduction*. A *β-redex* $(\lambda X s)t$ β-reduces to $[t/X]s$. An *η-redex* $\lambda X(sX)$ where variable X is not free in s, η-reduces to s. We write $s =_\beta t$ to mean s can be converted to t by a series of β-reductions and expansions. Similarly, $s =_{\beta\eta} t$ means s can be converted to t using both β and η.

For each simply typed λ-term s there is a unique *β-normal form* (denoted $s{\downarrow_\beta}$) and a unique $\beta\eta$-normal form (denoted $s{\downarrow_{\beta\eta}}$). From this fact we know $s \equiv_\beta t$ ($s \equiv_{\beta\eta} t$) if and only if $s{\downarrow_\beta} \equiv t{\downarrow_\beta}$ ($s{\downarrow_{\beta\eta}} \equiv t{\downarrow_{\beta\eta}}$).

Remember, that formulas are defined as terms of type o. A *non-atomic formula* is any formula whose β-normal form is of the form (cs) or $((cs)t)$ where c is a primitive logical connective. An *atomic formula* is any other formula.

The semantics of HOL is well understood and thoroughly documented in the literature. Here we briefly recapitulate some essential aspects. A more detailed overview can be found in Benzmüller and Miller [20].

Gödel's incompleteness theorem [42] can be extended directly to HOL since second-order quantification can be used to define Peano arithmetic: that is, there is a "true" formula of HOL (or any extension of it) that is not provable. The notion of truth here, however, is that arising from what is called the *standard model* of HOL in which a functional type, say, $\alpha \to \beta$, contains *all* functions from the type α to the type β. Moreover, the type o is assumed to contain exactly two truth values, namely *truth* and *falsehood*.

Henkin [44] introduced a broader notion of *general model* in which a type contains "enough" functions but not necessarily all functions. Henkin then showed soundness and completeness. More precisely, he showed that provability in HOL coincides with truth in all general models (the standard one as well as the non-standard ones).

Andrews [4] provided an improvement on Henkin's definition of general models by replacing the notion that there be enough functions to provide denotations for all formulas of HOL with a more direct means to define general models based on combinatory logic. Andrews [3] points out that Henkin's definition of general model technically was in error since his definition of general models admitted models in which the axiom of functional extensionality does not hold. Andrews then showed that there is a rather direct way to fix that problem by shifting the underlying logical connectives away from the usual Boolean connectives and quantifiers for a type-indexed family of connectives $\{Q_{\tau\to\tau\to o}\}_\tau$ in which $Q_{\tau\to\tau\to o}$ denotes equality at type τ. An indirect solution, which we also employ here, is to presuppose the presence of the identity relations in all domains $D_{\alpha\to\alpha\to o}$, which ensures the existence of unit sets $\{a\} \in D_{\alpha\to o}$ for all elements $a \in D_\alpha$. The existence of these unit sets in turn ensures that Leibniz equality indeed denotes the intended (fully extensional) identity relation.

Thus, Henkin models with Andrews' correction are fully extensional, i.e., they validate the functional and Boolean extensionality axioms. The Boolean extensionality axiom (abbreviated as $\mathcal{B}_o$) is given as

$$\forall A_o \forall B_o (A \longleftrightarrow B) \to A \doteq^o B$$

The infinitely many functional extensionality axioms (abbreviated as $\mathcal{F}_{\alpha\beta}$) are parameterized over $\alpha, \beta \in T$. They are given as

$$\forall F_{\alpha\to\beta} \forall G_{\alpha\to\beta} (\forall X_\alpha FX \doteq^\beta GX) \to F \doteq^{\alpha\to\beta} G$$

The construction of non-functional models has been pioneered by Andrews [2]. In Andrews's so-called v-complexes, which are based on Schütte's semi-valuation method [59], both the functional and the Boolean extensionality principles fail. Assuming β-equality, functional extensionality splits into two weaker and independent principles η ($F \doteq \lambda XFX$, if X is not free in term F) and ξ (from $\forall X.F \doteq G$ infer $\lambda XF \doteq \lambda XG$, where X may occur free in F and G). Conversely, $\beta\eta$-conversion, which is built-in in many modern implementations of HOL, together with ξ implies functional extensionality. Boolean extensionality, however, is independent of any of these principles. A whole landscape of respective notions of models structures between Andrews's v-complexes and Henkin semantics that further illustrate and clarify the above connections has been developed by Benzmüller, Brown and Kohlhase [14, 26, 6], and an alternative development and discussion has been contributed by Muskens [50].

The semantics of choice for the remainder of this work is Henkin semantics, i.e., we work with Henkin's general models. Henkin models (and standard models) are introduced next. We start out with introducing frame structures.

A *frame* D is a collection $\{D_\alpha\}_{\alpha\in \mathrm{T}}$ of nonempty sets D_α, such that $D_o = \{T, F\}$ (for truth and falsehood). The $D_{\alpha\to\beta}$ are collections of functions mapping D_α into D_β.

A *model* for HOL is a tuple $M = \langle D, I\rangle$, where D is a frame, and I is a family of typed interpretation functions mapping constant symbols $p_\alpha \in C_\alpha$ to appropriate elements of D_α, called the *denotation of* p_α (the logical connectives $\neg$, $\vee$, and $\forall$ are always given the standard denotations, cf. below). Moreover, we assume that the domains $D_{\alpha\to\alpha\to o}$ contain the respective identity relations.
Variable assignments are a technical aid for the subsequent definition of an interpretation function $\|.\|^{M,g}$ for HOL terms. This interpretation function is parametric over a model M and a variable assignment g.

A *variable assignment* g maps variables X_α to elements in D_α. $g[d/W]$ denotes the assignment that is identical to g, except for variable W, which is now mapped to d.

The *denotation* $\|s_\alpha\|^{M,g}$ of an HOL term s_α on a model $M = \langle D, I\rangle$ under assignment g is an element $d \in D_\alpha$ defined in the following way:

1. $\|p_\alpha\|^{M,g} = I(p_\alpha)$
2. $\|X_\alpha\|^{M,g} = g(X_\alpha)$
3. $\|(s_{\alpha\to\beta}\, t_\alpha)_\beta\|^{M,g} = \|s_{\alpha\to\beta}\|^{M,g}(\|t_\alpha\|^{M,g})$
4. $\|(\lambda X_\alpha\text{.}\, s_\beta)_{\alpha\to\beta}\|^{M,g}$ = the function f from D_α to D_β such that $f(d) = \|s_\beta\|^{M,g[d/X_\alpha]}$ for all $d \in D_\alpha$
5. $\|(\neg_{o\to o}\, s_o)_o\|^{M,g} = T$ if and only if $\|s_o\|^{M,g} = F$
6. $\|((\vee_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ if and only if $\|s_o\|^{M,g} = T$ or $\|t_o\|^{M,g} = T$
7. $\|(\forall_{(\alpha\to o)\to o}(\lambda X_\alpha\text{.}\, s_o))_o\|^{M,g} = T$ if and only if for all $d \in D_\alpha$ we have $\|s_o\|^{M,g[d/X_\alpha]} = T$

A model $M = \langle D, I\rangle$ is called a *standard model* if and only if for all $\alpha, \beta \in T$ we have $D_{\alpha\to\beta} = \{f \mid f : D_\alpha \longrightarrow D_\beta\}$. In a *Henkin model (general model)* function spaces are not necessarily full. Instead it is only required that $D_{\alpha\to\beta} \subseteq \{f \mid f : D_\alpha \longrightarrow D_\beta\}$ (for all $\alpha, \beta \in T$) and that the valuation function $\|\cdot\|^{M,g}$ from above is total (i.e., every term denotes). Any standard model is obviously also a Henkin model.

Truth in a model, validity in a model M and general validity are defined as usual: An HOL formula s_o is *true* in model M for world w under assignment g if and only if $\|s_o\|^{M,g} = T$; this is also denoted by $M, g \models^{\mathrm{HOL}} s_o$. An HOL formula s_o is called *valid* in M, which is denoted by $M \models^{\mathrm{HOL}} s_o$, if and only if $M, g \models^{\mathrm{HOL}} s_o$ for all assignments g. Moreover, a formula s_o is called *valid*, which we denote by $\models^{\mathrm{HOL}} s_o$, if and only if s_o is valid for all M. Finally, we define $S \models^{\mathrm{HOL}} s_o$ for a set of HOL formulas S if and only if $M \models^{\mathrm{HOL}} s_o$ for all models M with $M \models^{\mathrm{HOL}} t_o$ for all $t_o \in S$.

3.1 Cut-free Sequent Calculi for HOL

Cut-free sequent calculi for elementary type theory and fragments of it have been studied by Takeuti [66], Schütte [59], Tait [64], Takahashi [65], Prawitz [55], and Girard [41]. Andrews [2] used the *abstract consistency principle* of Smullyan [60] in order to give a proof of the completeness of resolution in elementary type theory. Takeuti [68] presented a cut-free sequent calculus with extensionality that is complete for Henkin models. The abstract consistency proof technique, as used by Andrews, has been further extended and applied in [47,6,26,14–16,29] to obtain cut-elimination results for different systems between elementary type theory and HOL.

We here present the cut-free, sound and complete, one-sided sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$ for HOL (without choice) by Benzmüller, Brown and Kohlhase [16]. In the context of this work, a sequent is a finite set Δ of β-normal closed formulas. A sequent calculus $\mathcal{G}$ provides an inductive definition for when $\vdash^{\mathcal{G}} \Delta$ holds. A sequent calculus rule

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\, r$$

is *admissible* in $\mathcal{G}$ if $\vdash^{\mathcal{G}} \Delta$ holds whenever $\vdash^{\mathcal{G}} \Delta_i$ for all $1 \leq i \leq n$.

Definition 1 (Sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$) Let Δ and Δ' be finite sets of β-normal closed formulas of HOL and let Δ, s denote the set $\Delta \cup \{s\}$. The sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$ comprises the following rules:

Basic Rules

$$\frac{\Delta, s}{\Delta, \neg\neg s}\,\mathcal{G}(\neg) \qquad \frac{\Delta, \neg s \quad \Delta, \neg t}{\Delta, \neg(s \vee t)}\,\mathcal{G}(\vee_-) \qquad \frac{\Delta, s, t}{\Delta, (s \vee t)}\,\mathcal{G}(\vee_+)$$

$$\frac{\Delta, \neg\, (sl){\downarrow_\beta} \quad l_\alpha \text{ closed term}}{\Delta, \neg \Pi^\alpha s}\,\mathcal{G}(\Pi^l_-) \qquad \frac{\Delta, (sc){\downarrow_\beta} \quad c_\delta \text{ new symbol}}{\Delta, \Pi^\alpha s}\,\mathcal{G}(\Pi^c_+)$$

Initialization

$$\frac{s \text{ atomic (and } \beta\text{-normal)}}{\Delta, s, \neg s}\,\mathcal{G}(init)$$

$$\frac{\Delta, (s \doteq^o t) \quad s,t \text{ atomic}}{\Delta, \neg s, t}\,\mathcal{G}(Init^{\doteq})$$

Extensionality

$$\frac{\Delta, (\forall X_\alpha sX \doteq^\beta tX)\Big\downarrow_\beta}{\Delta, (s \doteq^{\alpha\to\beta} t)}\,\mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta, \neg s, t \quad \Delta, \neg t, s}{\Delta, (s \doteq^o t)}\,\mathcal{G}(\mathfrak{b})$$

Decomposition

$$\frac{\Delta, (s^1 \doteq^{\alpha_1} t^1) \ \cdots \ \Delta, (s^n \doteq^{\alpha_n} t^n) \quad n \geq 1, \beta \in \{o, \iota\}, h_{\overline{\alpha^n}\to\beta} \in \Sigma}{\Delta, (h\overline{s^n} \doteq^\beta h\overline{t^n})}\,\mathcal{G}(d)$$

Theorem proving in $\mathcal{G}_{\beta\mathfrak{fb}}$ works as follows:[1] In order to prove that a (closed) conjecture formula c logically follows from a (possibly empty) set of (closed) axioms $\{a^1, \ldots, a^n\}$, we start from the initial sequent $\Delta := \{c, \neg a^1, \ldots, \neg a^n\}$ and reason backwards by applying the respective calculus rules. We are done, if all branches of the proof tree can be closed by an application of the $\mathcal{G}(init)$ rule. In this case $\vdash^{\mathcal{G}_{\beta\mathfrak{fb}}} \Delta := \{c, \neg a^1, \ldots, \neg a^n\}$ holds, which means that the conjecture c logically follows from the axioms $a^1, \ldots, a^n$ within calculus $\mathcal{G}_{\beta\mathfrak{fb}}$.

Soundness and completeness of $\mathcal{G}_{\beta\mathfrak{fb}}$ for HOL with Henkin semantics has been established in [16].

Theorem 1 (Soundness and Completeness $\mathcal{G}_{\beta\mathfrak{fb}}$ for HOL)

$$\models^{HOL} s \text{ if and only if } \vdash^{\mathcal{G}_{\beta\mathfrak{fb}}} \{s\}$$

More generally, $\{a^1, \ldots, a^n\} \models^{HOL} s$ *if and only if* $\vdash^{\mathcal{G}_{\beta\mathfrak{fb}}} \{s, \neg a^1, \ldots, \neg a^n\}$.

Note that rule $\mathcal{G}(cut)$

$$\frac{\Delta, s \quad \Delta, \neg s}{\Delta}\ \mathcal{G}(cut)$$

is not available in $\mathcal{G}_{\beta\mathfrak{fb}}$. Hence, cut-elimination holds [16].

Theorem 2 (Cut-Elimination) *The rule* $\mathcal{G}(cut)$ *is admissible in* $\mathcal{G}_{\beta\mathfrak{fb}}$.

In spite of their cut-freeness, both calculi are obviously only mildly suited for automation. One reason is that they are blindly guessing instantiations l in rule $\mathcal{G}(\Pi_-^l)$. Another reason is that the treatment of equality in both calculi relies on Leibniz equality $\doteq$. Support for primitive equality is not provided. The problem with Leibniz equality (or other forms of defined equality) is that it threatens cut-freeness of the calculi by allowing for simulations (admissibility) of the cut rule. The problem of cut-simulation, which poses a thread to effective proof automation, analogously applies to a wide range of prominent other HOL axioms. The issue is addressed in more depth in the next subsection.

3.2 Cut-Simulation in HOL

It is exemplarily illustrated why Leibniz equality implies cut-simulation. Assume we want to study in $\mathcal{G}_{\beta\mathfrak{fb}}$ whether a conjecture c logically follows from an equality axiom $l = r$ (where l and r are some arbitrary closed terms of type α). Since primitive equality is not available we formalize the axiom as $l \doteq^{\alpha} r$ and initialize the proof process with sequent $\Delta := \{c, \neg(l \doteq^{\alpha} r)\}$, that is, with $\Delta := \{c, \neg\Pi(\lambda P_{\alpha\to o}(\neg Pl \vee Pr))$.

[1] It is a recommended exercise to verify that η-equality, and hence $\beta\eta$-equality, is implied by the rules of $\mathcal{G}_{\beta\mathfrak{fb}}$.

Now consider the following derivation, where s is an arbitrary (cut) formula:

$$\dfrac{\dfrac{\dfrac{\Delta, s}{\Delta, \neg\neg s}\,\mathcal{G}(\neg) \qquad \Delta, \neg s}{\Delta, \neg(\neg s \lor s)}\,\mathcal{G}(\lor_-)}{\Delta, \neg\Pi(\lambda P_{\alpha\to o}(\neg Pl \lor Pr))}\,\mathcal{G}(\Pi_-^{\lambda X s})$$

It is easy to see that this derivation introduces a cut on formula s; in the left branch s occurs positively and in the right branch negatively.

Cut-simulation is also enabled by the functional and Boolean extensionality axioms, cf. $\mathcal{F}_{\alpha\beta}$ and $\mathcal{B}_o$ above. Instead of the extensionality rules $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\mathfrak{b})$, as provided in calculus $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$, we could alternatively postulate the validity of these axioms. For this we could replace the rules $\mathcal{G}(\mathfrak{f})$ and $\mathcal{G}(\mathfrak{b})$ in $\mathcal{G}_\beta$ by the following axiomatic extensionality rules $\mathcal{G}(\mathcal{F}_{\alpha\beta})$ and $\mathcal{G}(\mathcal{B}_o)$:

$$\dfrac{\Delta, \neg\mathcal{F}_{\alpha\beta} \quad \alpha\to\beta\in T}{\Delta}\,\mathcal{G}(\mathcal{F}_{\alpha\beta}) \qquad\qquad \dfrac{\Delta, \neg\mathcal{B}_o}{\Delta}\,\mathcal{G}(\mathcal{B}_o)$$

This calculus is still Henkin complete (even if rules $\mathcal{G}(d)$ and $\mathcal{G}(Init^{\doteq})$ are additionally removed) [16]. However, the modified calculus suffers severely from cut-simulation. For axiom $\mathcal{B}_o$ this is illustrated by the following derivation (a_o is new constant symbol):

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{derivable in 7 steps}\\ \vdots\\ \Delta, a \longleftrightarrow a\end{array}}{\Delta, \neg\neg(a \longleftrightarrow a)}\,\mathcal{G}(\neg) \qquad \begin{array}{c}\Delta, s \quad \Delta, \neg s\\ \vdots\ \text{derivable in 3 steps, see above}\\ \Delta, \neg(a \doteq^o a)\end{array}}{\Delta, \neg(\neg(a \longleftrightarrow a) \lor a \doteq^o a)}\,\mathcal{G}(\lor_-)}{\Delta, \neg\mathcal{B}_o}\,2\times\mathcal{G}(\Pi_-^a)$$

The left branch is closed and on the right branch an arbitrary cut formula s is introduced. A similar derivation is enabled with axiom $\mathcal{F}_{\alpha\beta}$ (b_α is new constant symbol):

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{derivable in 3 steps}\\ \vdots\\ \Delta, fb \doteq^\beta fb\end{array}}{\Delta, (\forall X_\alpha fX \doteq^\beta fX)}\,\mathcal{G}(\Pi_+^b)}{\Delta, \neg\neg\forall X_\alpha fX \doteq^\beta fX}\,\mathcal{G}(\neg) \qquad \begin{array}{c}\Delta, s \quad \Delta, \neg s\\ \vdots\ \text{derivable in 3 steps}\\ \Delta, \neg(f \doteq^{\alpha\to\beta} f)\end{array}}{\Delta, \neg(\neg(\forall X_\alpha fX \doteq^\beta fX) \lor f \doteq^{\alpha\to\beta} f)}\,\mathcal{G}(\lor_-)}{\Delta, \neg\mathcal{F}_{\alpha\beta}}\,2\times\mathcal{G}(\Pi_-^f)$$

In all cut-simulations above we have exploited the fact that predicate variables may be instantiated with terms that introduce arbitrary new formulas s. At these points the subformula property breaks. At the same time this offers the opportunity to mimic cut-introductions by appropriately selecting such instantiations for predicate variables. In addition to Leibniz equations and the Boolean and functional extensionality axioms, cut-simulations are analogously enabled by many prominent other axioms, including excluded middle, description, choice, comprehension, and induction. We may thus call these axioms cut-strong. More details on such cut-strong axioms are provided in previous work [16].[2]

Cut-simulations have in fact been extensively used in literature. For example, Takeuti showed that a conjecture of Gödel could be proved without cut by using the induction principle instead [67]; McDowell and Miller [48] illustrate how the induction rule can be used to hide the cut rule; and Schütte [59] used excluded middle to similarly mask the cut rule.

For the development of automated proof procedures for HOL we thus learn an important lesson, namely that cut-elimination and cut-simulation should always be considered in combination: a pure cut-elimination result may indeed mean little if at the same time axioms are assumed that support effective cut-simulation. The challenge is to develop cut-free calculi for HOL that also try to avoid the pitfall of cut-simulations (as far as this is possible in given context).

Church's use of the λ-calculus to build comprehension principles into the language can therefore be seen as a first step in the program to eliminate cut-strong axioms. Significant progress in the automation of HOL in existing prover implementations has been achieved after providing calculus level support for extensionality and also choice (avoiding cut-simulation effects). Respective extensionality rules have been provided for resolution [6,7], expansion and sequent calculi [26,27], and tableaux [29]. Similarly, choice rules have been proposed for the various settings: sequent calculus [49], tableaux [5] and resolution [24].

The calculi as employed by automated theorem provers Leo-II [23] and Satallax [28] actually share significant conceptual similarities with the sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$, in particular, regarding the handling of extensionality, and they implement various other means to avoid cut-simulations.

4 Cut-Elimination via Semantic Embedding

With sequent calculus $\mathcal{G}_{\beta\mathfrak{fb}}$ an example of a cut-free calculus for HOL has been provided. Next, QCL is modelled as a proper fragment of HOL with Henkin semantics. This way we obtain a cut-elimination result for QCL for free.

[2] Obviously, any universally quantified predicate variable (occurring negatively in the above approach) is a possible source for cut-simulation. The challenge thus is to avoid those predicate variables as far as possible. An axiomatic approach based on cut-strong axioms, as proposed by several authors including e.g. [45,46], is therefore hardly a suitable option for the automation of HOL.

4.1 Modeling QCL as a fragment of HOL

Regarding the particular choice of HOL, we here assume a set of basic types $BT = \{o, i, u\}$, where o denotes the type of Booleans as before. Without loss of generality, i is now identified with a (non-empty) set of worlds and the additional base type u with a (non-empty) domain of individuals. The particular choice of whether u or i is identified with individuals, respectively worlds, is irrelevant and could be reversed. The selection here is motivated by the idea to stay as close as possible with the choices made in previous work.

QCL formulas are now 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. Type $i \to o$ is abbreviated as τ in the remainder.

The mapping $\lfloor\cdot\rfloor$ translates QCL formulas φ into HOL terms $\lfloor\varphi\rfloor$ of type τ. The mapping is recursively defined:

$$\begin{aligned}
\lfloor P \rfloor &= P_\tau \\
\lfloor k(X^1, \ldots, X^n) \rfloor &= k_{u^n \to \tau}\, X^1_u \ldots X^n_u \\
\lfloor \neg\varphi \rfloor &= \neg_\tau \lfloor\varphi\rfloor \\
\lfloor \varphi \vee \psi \rfloor &= \vee_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor \\
\lfloor \varphi \Rightarrow \psi \rfloor &= \Rightarrow_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor \\
\lfloor \forall^{co} X \varphi \rfloor &= \Pi^{co}_{(u\to\tau)\to\tau}\, \lambda X_u \lfloor\varphi\rfloor \\
\lfloor \forall^{va} X \varphi \rfloor &= \Pi^{va}_{(u\to\tau)\to\tau}\, \lambda X_u \lfloor\varphi\rfloor \\
\lfloor \forall P \varphi \rfloor &= \Pi_{(\tau\to\tau)\to\tau}\, \lambda P_\tau \lfloor\varphi\rfloor
\end{aligned}$$

P_τ is a variable of type τ and $X^1_u, \ldots, X^n_u$ are variables of type u. $k_{u^n\to\tau}$ (for $n \geq 0$) is a constant symbol of type $\underbrace{u \to \ldots \to u}_{n} \to \tau$. $\neg_\tau$, $\vee_{\tau\to\tau\to\tau}$, $\Rightarrow_{\tau\to\tau\to\tau}$, $\Pi^{co,va}_{(u\to\tau)\to\tau}$ and $\Pi_{(\tau\to\tau)\to\tau}$ realize the QCL connectives in HOL. They abbreviate the following HOL terms:[3]

$$\begin{aligned}
\neg_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \neg(A\,X) \\
\vee_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i (A\,X \vee B\,X) \\
\Rightarrow_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i \forall V_i (f\,X\,A\,V \to B\,V) \\
\Pi^{co}_{(u\to\tau)\to\tau} &= \lambda Q_{u\to\tau} \lambda V_i \forall X_u (Q\,X\,V) \\
\Pi^{va}_{(u\to\tau)\to\tau} &= \lambda Q_{u\to\tau} \lambda V_i \forall X_u (eiw\,V\,X \to Q\,X\,V) \\
\Pi_{(\tau\to\tau)\to\tau} &= \lambda R_{\tau\to\tau} \lambda V_i \forall P_\tau (R\,P\,V)
\end{aligned}$$

Constant symbol f in the mapping of $\Rightarrow$ is of type $i \to \tau \to \tau$. It realizes the selection function. Constant symbol eiw (for 'exists in world'), which is of type $(\tau \to u) \to \tau$, is associated with the varying domains. The interpretations of f and eiw are chosen appropriately below. Moreover, for the varying domains a non-emptiness axiom is postulated:

$$\forall W_i \exists X_u (eiw\,W\,X) \qquad \text{(NE)}$$

[3] Note the predicate argument A of f in the term for $\Rightarrow_{\tau\to\tau\to\tau}$ and the second-order quantifier $\forall P_\tau$ in the term for $\Pi_{(\tau\to\tau)\to\tau}$. FOL encodings of both constructs, if feasible, will be less natural.

The above mapping induces mappings $\lfloor \text{IV} \rfloor$, $\lfloor \text{PV} \rfloor$ and $\lfloor \text{SYM} \rfloor$ of the sets IV, PV and 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 previous work [17,19,21,22], we define $\text{vld}_{\tau \to o} = \lambda A_\tau \forall S_i (A\, S)$. With this definition, validity of a QCL formula φ in QCK corresponds to the validity of $(\text{vld}\, \lfloor \varphi \rfloor)$ in HOL, and vice versa.

To prove the soundness and completeness of the above embedding, a mapping from QCL models into Henkin models is employed [10]. This mapping utilizes a corresponding mapping of QCL variable assignments into HOL variable assignments.

Let $g = (g^i : \text{IV} \mapsto D,\, g^p : \text{PV} \mapsto Q)$ be a variable assignment for QCL. The *corresponding variable assignment* $\lfloor g \rfloor = (\lfloor g^i \rfloor : \lfloor \text{IV} \rfloor \mapsto D,\, \lfloor g^p \rfloor : \lfloor \text{PV} \rfloor \mapsto Q)$ for HOL is defined such that $\lfloor g \rfloor (X_u) = \lfloor g \rfloor (\lfloor X \rfloor) = g(X)$ and $\lfloor g \rfloor (P_\tau) = \lfloor g \rfloor (\lfloor P \rfloor) = g(P)$ for all $X_u \in \lfloor \text{IV} \rfloor$ and $P_\tau \in \lfloor \text{PV} \rfloor$. Finally, $\lfloor g \rfloor$ 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 $\alpha \neq u, \tau$.

Definition 2 (Henkin model H^M) Given a QCL model $M = \langle S, f, D, D', Q, I \rangle$. The Henkin model $H^M = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ for 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 $\lfloor g^i \rfloor$), D_τ is chosen as the set of sets of possible worlds Q (cf. definition of $\lfloor g^p \rfloor$)[4], 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 $D_{\alpha \to \beta}$ the rule that every term must have a denotation must be obeyed, in particular, it is required that $D_{u^n \to \tau}$, $D_{i \to \tau \to \tau}$ and $D_{i \to u \to o}$ contain the elements $I k_{u^n \to \tau}$, $I f_{i \to \tau \to \tau}$ and $I eiw_{i \to u \to o}$ as characterized next. Interpretation I is constructed as follows:

1. Let $k_{u^n \to \tau} = \lfloor k \rfloor$ for n-ary $k \in \text{SYM}$ $(n \geq 0)$ and let $X^i_u = \lfloor X^i \rfloor$ for $X^i \in \text{IV}$, $i = 1, \ldots, n$. $I k_{u^n \to \tau} \in D_{u^n \to \tau}$ is chosen such that $(I\, k_{u^n \to \tau})(\lfloor g \rfloor (X^1_u), \ldots, \lfloor g \rfloor (X^n_u), w) = T$ for all worlds $w \in D_i$ with $M, g, w \models k(X^1, \ldots, X^n)$, i.e., if $\langle g^i(X^1), \ldots, g^i(X^n) \rangle \in I(k, w)$. Otherwise, $(I\, k_{u^n \to \tau})(\lfloor g \rfloor (X^1_u), \ldots, \lfloor g \rfloor (X^n_u), w) = F$.
2. $I\, f_{i \to \tau \to \tau} \in D_{i \to \tau \to \tau}$ is chosen such that $(I f_{i \to \tau \to \tau})(s, q, t) = T$ for all worlds $s, t \in D_i$ and $q \in D_\tau$ with $t \in f(s, \{x \in S \mid q(x) = T\})$ in M. Otherwise, $(I f_{i \to \tau \to \tau})(s, q, t) = F$.
3. $I\, eiw_{i \to u \to o} \in D_{i \to u \to o}$ is chosen such that $(I\, eiw_{i \to u \to o})(s, d) = T$ for individuals $d \in D'(s)$ in M. Otherwise, $(I\, eiw_{i \to u \to o})(s, d) = F$.
4. For all other constants c_α, choose $I c_\alpha$ arbitrary.[5]

It is not hard to verify that H^M is a Henkin model.[6]

[4] Sets are identified with their characteristic functions.

[5] In fact, it may be safely assumed that there are no other typed constant symbols given, except for the symbols $f_{i \to \tau \to \tau}$, $eiw_{i \to u \to o}$, $k_{u^n \to \tau}$, and the logical connectives.

[6] In H^M we have merely fixed $D_i = S$, $D_u = D$ and $D_\tau = Q$, and the interpretations of $k_{u^n \to \tau}$, $f_{i \to \tau \to \tau}$ and $eiw_{i \to u \to o}$. These choices are not in conflict with any of the requirements

Lemma 1 *Let H^M be a Henkin model for a QCL model M which validates axiom NE. For all quantified conditional logic formulas δ, variable assignments g and worlds s it holds: $M, g, s \models \delta$ if and only if $\| \lfloor \delta \rfloor \, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = T$.*

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 [17], Lemma 1.

If $\delta = \forall^{va} X\varphi$, by definition, it holds $M, g, s \models \forall^{va} X\varphi$ if and only if for all $d \in D'(s)$ we have $M, ([d/X_u]g^i, g^p), s \models \varphi$. By induction the latter condition is equivalent to: for all $d \in D'(s)$ holds $\| \lfloor \varphi \rfloor \, S_i \|^{H^M, \lfloor ([d/X_u]g^i, g^p) \rfloor [s/S_i]} = \| \lfloor \varphi \rfloor \, S_i \|^{H^M, (\lfloor g \rfloor [s/S_i])[d/X_u]} = T$. Due to the choice of $I\, eiw$ in H^M, the non-emptiness condition NE and definition of $\|.\|$ this is equivalent to: for all $d \in D$ it holds $\| eiw\, S_i\, X_u \rightarrow \lfloor \varphi \rfloor \, S_i \|^{H^M, \lfloor g \rfloor [s/S_i][d/X_u]}$. Hence, by definition of $\|.\|$ we have that $\| \Pi_{(u \to o) \to o} \lambda X_u (eiw\, S_i\, X_u \rightarrow \lfloor \varphi \rfloor \, S_i) \|^{H^M, \lfloor g \rfloor [s/S_i]} =_{\beta\eta} \| (\lambda W_i\, \Pi_{(u \to o) \to o}\, \lambda X_u (eiw\, W_i\, X_u \rightarrow \lfloor \varphi \rfloor \, W_i))\, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = T$ By definition of $\Pi^{va}_{(u \to \tau) \to \tau}$, definition of $\lfloor . \rfloor$ and definition of $\|.\|$ we finally get $\| (\Pi^{va}_{(u \to \tau) \to \tau}\, \lambda X_u \lfloor \varphi \rfloor)\, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = \| \lfloor \forall^{va} X\varphi \rfloor \, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = T$

The cases for $\delta = \forall^{co} X\varphi$ and $\delta = \forall P\varphi$ are similar (actually simpler).

Theorem 3 (Soundness and Completeness of the Embedding)

$$\models^{QCL} \varphi \text{ if and only if } \{NE\} \models^{HOL} vld\, \lfloor \varphi \rfloor$$

Proof (Soundness, $\leftarrow$) The proof is by contraposition. Assume $\not\models^{QCL} \varphi$, i.e, there is a QCL model $M = \langle S, f, D, D', Q, I \rangle$, an assignment g and a world $s \in S$, such that $M, g, s \not\models \varphi$. By Lemma 1 it holds that $\| \lfloor \varphi \rfloor \, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = F$ in Henkin model $H^M = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ for M. Furthermore, $H^M \models^{HOL} \{NE\}$. Thus, by definition of $\|.\|$, definition of vld and since $\forall S_i (\lfloor \varphi \rfloor \, S_i) =_{\beta\eta} vld\, \lfloor \varphi \rfloor$ it holds that $\| \forall S_i (\lfloor \varphi \rfloor \, S_i) \|^{H^M, \lfloor g \rfloor} = \| vld\, \lfloor \varphi \rfloor \|^{H^M, \lfloor g \rfloor} = F$ Hence, $H^M \not\models^{HOL} vld\, \lfloor \varphi \rfloor$, and thus $\{NE\} \not\models^{HOL} vld\, \lfloor \varphi \rfloor$.

(Completeness, $\rightarrow$) The proof is again by contraposition. Assume $\{NE\} \not\models^{HOL} vld\, \lfloor \varphi \rfloor$, i.e., there is a Henkin model $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ and an assignment ψ such that $H \models^{HOL} \{NE\}$ and $\| vld\, \lfloor \varphi \rfloor \|^{H, \psi} = F$. Without loss of generality it can be assumed that Henkin model H is in fact a Henkin model H^M for a corresponding QCL model M and that $\psi = \lfloor g \rfloor$ for a corresponding QCL variable assignment g. By definition of $\|.\|$ and since $vld\, \lfloor \varphi \rfloor =_{\beta\eta} \forall S_i (\lfloor \varphi \rfloor \, S_i)$ it holds $\| \forall S_i (\lfloor \varphi \rfloor \, S_i) \|^{H^M, \lfloor g \rfloor} = F$, and hence, by definition of vld, $\| \lfloor \varphi \rfloor \, S_i \|^{H^M, \lfloor g \rfloor [s/S_i]} = F$ for some $s \in D$. By Lemma 1 $M, g, s \not\models \varphi$, and hence $\not\models^{QCL} \varphi$.

---

regarding frames and interpretations. The existence of a valuation function V for an HOL interpretation crucially depends on how sparse the function spaces have been chosen in frame $\{D_\alpha\}_{\alpha \in T}$. Andrews [3] discusses criteria that are sufficient to ensure the existence of a valuation function; they require that certain λ-abstractions have denotations in frame $\{D_\alpha\}_{\alpha \in T}$. Since it is explicitly required that every term denotes and since Q has been appropriately constrained in QCL models (and hence $D_{i \to o}$ in H^M) these requirements are met.

This shows that QCL is a natural fragment of HOL.
Combining Theorem 3 with Theorem 1 we obtain:

Theorem 4 (Soundness and Completeness of $\mathcal{G}_{\beta\mathfrak{fb}}$ for QCL)

$$\models^{QCL} \varphi \textit{ if and only if } \vdash^{\mathcal{G}_{\beta\mathfrak{fb}}} \{vld\lfloor\varphi\rfloor, \neg NE\}$$

Since $\mathcal{G}_{\beta\mathfrak{fb}}$ is cut-free (Theorem 2), we thus obtain a cut-elimination result for QCL for free.

Corollary 1 (Cut-Elimination for QCL)

$$\models^{QCL} \varphi \textit{ if and only if } \vdash^{\mathcal{G}_{\beta\mathfrak{fb}}}_{\textit{cut-free}} \{vld\lfloor\varphi\rfloor, \neg NE\}$$

The above result holds for base logic QCK. Further prominent QCLs can be handled by simply postulating respective (combinations of) axioms, cf. the axioms ID, MP, CS, CEM, AC, RT CV, and CA in Figure 1. Postulating these axioms is easily possible in our framework since our notion of QCL provides quantification over propositional variables.

Obviously, our cut-elimination result still applies to the resulting QCL variants. The above corollary also entails a cut-elimination result for quantified modal logic, since QCL subsumes normal modal logic as noted before.

However, we need to point again to the subtle issue of cut-simulation. In particular, note that all axioms from Figure 1 introduce predicate variables. Such predicate variables have been the source of the cut-simulation effects as described before; cf. step $\mathcal{G}(\Pi^{\lambda X s}_{-})$ on page 11.

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

Fig. 1 Conditional logic axioms and semantic conditions

In some cases, however, the semantical conditions associated with the axioms can be postulated instead in order to circumvent the effect. This is in fact possible for many prominent modal logic axioms. For example, the corresponding semantical condition for modal axiom T: $\forall\varphi(\Box\varphi \supset \varphi)$ is $\forall x(rxx)$

(where constant r denotes the associated accessibility relation). Obviously, the latter reflexivity axiom does not support cut-simulation and it should therefore be preferred. Unfortunately, the semantical condition associated with ID, $\forall A_\tau \forall W_i (f\, W A \subseteq A)$, still introduces predicate variable A_τ, and so do all other semantical conditions displayed in Figure 1. It remains future work to study cut-simulation for the QCL axioms and their associated semantical conditions more closely.

4.2 Other Logic Embeddings in HOL

Recent work has shown that many other challenging logics can be characterized as HOL fragments via semantic embeddings. The logics studied so far include modal logics, intuitionistic logic, security logics, hybrid logics, logics for time and space, and logics based on neighborhood semantics [19,21,22,8,9,18,17,25,69,53]. The studied fragments also comprise first-order and even higher-order extensions of non-classical logics, for which only little practical automation support has been available so far. Most importantly, however, combinations of embedded logics can be elegantly achieved in our approach. And, analogous to above, cut-elimination results for these embedded logics can be obtained "for free" by exploiting the results already achieved for HOL. Moreover, cut-simulation effects can often be avoided.

It is important to note that the embedding approach is not only of theoretical relevance. In fact, the approach has been employed in practice in combination with existing higher-order theorem provers [10,17,21,9,22,8]. Most importantly, we want to point to the very successful application of the approach for the verification and automation of Gödel's ontological argument [25]. In this work an embedding of higher-order modal logics in HOL has been utilized, and the HOL reasoners have in fact contributed some novel insights. The next subsection briefly illustrates how the embedding of QCL in HOL can be utilized in practice.

5 Utilizing the Embedding Approach in Practice

Figure 2 presents the HOL encoding of QCL connectives and axiom NE in THF syntax; THF is a concrete syntax for HOL [63]. We here introduce definitions for the logical connectives $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\equiv$, $\Rightarrow$, $\forall^{co}$, $\forall^{va}$ and $\forall$. Further definitions, for example, for $\bot$, $\top$ or the existential quantifiers $\exists^*$, can be easily added.

The content of Figure 2, if stored e.g. in a file `QCL.ax`, can be imported in other files to support practical experiments with reasoning in QCL. By employing the QCL connectives provided in `QCL.ax`, QCL statements can be directly expressed and automated with off-the-shelf automated theorem provers for HOL such as LEO-II and Satallax.[7] In other words, file `QCL.ax`

[7] Experiments with these and other reasoners for THF are supported online via Sutcliffe's SystemOnTPTP infrastructure [62]; cf. `www.tptp.org/cgi-bin/SystemOnTPTP`.

```
%--- file:QCL.ax
%--- type u for individuals
thf(u,type,(u:$tType)).
%--- reserved constant for selection function f
thf(f,type,(f:$i>($i>$o)>$i>$o)).
%--- exists in world (eiw) predicate for varying dom.; non-emptiness axiom NE
thf(eiw,type,(eiw:$i>u>$o)).
thf(ne,axiom,(![V:$i]:?[X:u]:(eiw@V@X))).
%--- negation, disjunction, conjunction, material implication, equivalence
thf(not,type,(not:($i>$o)>$i>$o)).
thf(or,type,(or:($i>$o)>($i>$o)>$i>$o)).
thf(and,type,(and:($i>$o)>($i>$o)>$i>$o)).
thf(impl,type,(impl:($i>$o)>($i>$o)>$i>$o)).
thf(equiv,type,(equiv:($i>$o)>($i>$o)>$i>$o)).
thf(not_def,definition,(not = (^[A:$i>$o,X:$i]:~(A@X)))).
thf(or_def,definition,(or = (^[A:$i>$o,B:$i>$o,X:$i]:((A@X)|(B@X))))).
thf(and_def,definition,(and = (^[A:$i>$o,B:$i>$o,X:$i]:((A@X)&(B@X))))).
thf(impl_def,definition,(impl = (^[A:$i>$o,B:$i>$o,X:$i]:((A@X)=>(B@X))))).
thf(equiv_def,definition,(equiv =(^[A:$i>$o,B:$i>$o,X:$i]:((A@X)<=>(B@X))))).
%--- conditionality
thf(cond,type,(cond:($i>$o)>($i>$o)>$i>$o)).
thf(cond_def,definition,(cond
 = (^[A:$i>$o,B:$i>$o,X:$i]:![W:$i]:((f@X@A@W)=>(B@W))))).
%--- quantification (constant dom., varying dom., prop.)
thf(all_co,type,(all_co: (u>$i>$o)>$i>$o)).
thf(all_va,type,(all_va:(u>$i>$o)>$i>$o)).
thf(all,type,(all:(($i>$o)>$i>$o)>$i>$o)).
thf(all_co_def,definition,(all_co = (^[A:u>$i>$o,W:$i]:![X:u]:(A@X@W)))).
thf(all_va_def,definition,(all_va
 = (^[A:u>$i>$o,W:$i]:![X:u]:((eiw@W@X)=>(A@X@W))))).
thf(all_def,definition,(all = (^[A:($i>$o)>$i>$o,W:$i]:![P:$i>$o]:(A@P@W)))).
%--- notion of validity of a conditional logic formula
thf(vld,type,(vld:($i>$o)>$o)).
thf(vld_def,definition,(vld = (^[A:$i>$o]:![S:$i]:(A@S)))).
```

Fig. 2 THF encoding of QCL. Some notes on THF: `$i` and `$o` represent the HOL base types i and o. `$i>$o` encodes a function (predicate) type. Function or predicate application as in $(eiw\, V\, X)$ is encoded as `((eiw@V)@X)` or simply as `(eiw@V@X)`, i.e., function application is represented by `@`. Universal quantification and λ-abstraction as in $\lambda A_{i \to o} \forall S_i (A\, S)$ are represented as in `^[A:$i>$o]:![S:$i]:(A@S)`; `?` is the existential quantifier, and $\neg, \vee, \wedge, and \Rightarrow$ (mat. impl.) are written as , `|`, `&`, and `=>`. Comments begin with `%`. Better formatted and easier readable presentations of the THF code in this paper can easily be generated with the TPTP tools [62] of Sutcliffe available at `www.tptp.org`; here the minimization of space has been of primary interest. More information on the THF syntax (and the associated infrastructure) is available elsewhere [63].

```
%--- file: RCK.p
include('QCL.ax').
%--- rule RCK (for n=2) is admissible
thf(rck_2,conjecture,(
 (vld@(all@^[P1:$i>$o]:(all@^[P2:$i>$o]:(all@^[Q:$i>$o]:
   (equiv@(and@P1@P2)@Q)))))
 =>
 (vld@(all@^[P0:$i>$o]:(all@^[P1:$i>$o]:(all@^[P2:$i>$o]:(all@^[Q:$i>$o]:
   (impl@(and@(cond@P0@P1)@(cond@P0@P2))@(cond@P0@Q))))))) )).
```

Fig. 3 THF encoding of the admissibility of rule RCK (for n=2).

turns automated theorem provers for HOL into automated theorem provers for QCL. Moreover, the theorem provers confirm in a few milliseconds that the definitions and axioms provided in this file are satisfiable.

An example proof problem is given in Figure 3, where the meta-logical theorem is formulated that rule RCK (for $n = 2$) is admissible in the embeddings approach. Note that quantification over propositional variables is exploited here.

The statement, stored in a file `RCK.p`, can be proved by the automated theorem provers LEO-II and Satallax in a few milliseconds on a standard notebook; further experiments have been reported in previous work [17, 10].

6 Conclusion

The embeddings approach bridges between the Tarski view of logics (for metalogic HOL) and the Kripke view (for the embedded source logics) and exploits the fact that well-known translations of logics, respectively, their Kripke-style semantical characterizations, can often be elegantly and directly formalized in HOL. As demonstrated in this article for quantified conditional logics, cut-elimination results can thus be very easily obtained for the embedded logics via reduction to existing cut-elimination results for HOL. However, as also alluded, in some cases the obtained cut-elimination results may be pointless due to cut-simulation effects.

Cut-elimination results, as exemplarily provided in this article, are only one prominent aspect to justify this research direction from a theoretical perspective. A more pragmatic motivation is that the embeddings approach enables existing higher-order automated theorem proving systems to be uniformly applied to reason *within* and also *about* embedded logics and their combinations, and respective practical evidence has already been provided in previous work. In particular, there is not a single other implementation of a reasoner for quantified conditional logic (or higher-order modal logic) available to date.

Acknowledgements I am grateful to several colleagues for fruitful discussions and contributions to earlier related work; this list of persons includes Chad Brown, Dov Gabbay, Valerio Genovese, Larry Paulson, Dale Miller, Nik Sultana, and Bruno Woltzenlogel Paleo. Moreover, I am grateful to Alexander Steen for proof reading this article.

References

1. Andrews, P.: Church's type theory. In: E.N. Zalta (ed.) The Stanford Encyclopedia of Philosophy, Spring 2014 edn., Stanford University (2014)
2. Andrews, P.B.: Resolution in type theory. Journal of Symbolic Logic **36**(3), 414–432 (1971)
3. Andrews, P.B.: General models and extensionality. Journal of Symbolic Logic **37**(2), 395–397 (1972)
4. Andrews, P.B.: General models, descriptions, and choice in type theory. Journal of Symbolic Logic **37**(2), 385–394 (1972)

5. Backes, J., Brown, C.E.: Analytic tableaux for higher-order logic with choice. Journal of Automated Reasoning **47**(4), 451–479 (2011)
6. Benzmüller, C.: Equality and extensionality in automated higher-order theorem proving. Ph.D. thesis, Saarland University (1999)
7. Benzmüller, C.: Extensional higher-order paramodulation and RUE-resolution. In: H. Ganzinger (ed.) Automated Deduction - CADE-16, 16th International Conference on Automated Deduction, Trento, Italy, July 7-10, 1999, Proceedings, no. 1632 in LNCS, pp. 399–413. Springer (1999). DOI 10.1007/3-540-48660-7_39.
8. Benzmüller, C.: Automating access control logic in simple type theory with LEO-II. In: D. Gritzalis, J. López (eds.) Emerging Challenges for Security, Privacy and Trust, 24th IFIP TC 11 International Information Security Conference, SEC 2009, Pafos, Cyprus, May 18-20, 2009. Proceedings, *IFIP*, vol. 297, pp. 387–398. Springer (2009). DOI 10.1007/978-3-642-01244-0_34.
9. Benzmüller, C.: Combining and automating classical and non-classical logics in classical higher-order logic. Annals of Mathematics and Artificial Intelligence (Special issue Computational logics in Multi-agent Systems (CLIMA XI)) **62**(1-2), 103–128 (2011). DOI 10.1007/s10472-011-9249-7.
10. Benzmüller, C.: Automating quantified conditional logics in HOL. In: F. Rossi (ed.) 23rd International Joint Conference on Artificial Intelligence (IJCAI-13), pp. 746–753. Beijing, China (2013).
11. Benzmüller, C.: Cut-free calculi for challenge logics in a lazy way. In: C.N. Clint van Alten Petr Cintula (ed.) Proceedings of the International Workshop on Algebraic Logic in Computer Science (2013).
12. Benzmüller, C.: A top-down approach to combining logics. In: Proc. of the 5th International Conference on Agents and Artificial Intelligence (ICAART), pp. 346–351. SciTePress Digital Library, Barcelona, Spain (2013). DOI 10.5220/0004324803460351.
13. Benzmüller, C.: Higher-order automated theorem provers. In: D. Delahaye, B. Woltzenlogel Paleo (eds.) All about Proofs, Proof for All, Mathematical Logic and Foundations, pp. 171–214. College Publications, London, UK (2015).
14. Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. Journal of Symbolic Logic **69**(4), 1027–1088 (2004). DOI 10.2178/jsl/1102022211.
15. Benzmüller, C., Brown, C., Kohlhase, M.: Cut elimination with xi-functionality. In: C. Benzmüller, C. Brown, J. Siekmann, R. Statman (eds.) Reasoning in Simple Type Theory — Festschrift in Honor of Peter B. Andrews on His 70th Birthday, Studies in Logic, Mathematical Logic and Foundations, pp. 84–100. College Publications (2008).
16. Benzmüller, C., Brown, C., Kohlhase, M.: Cut-simulation and impredicativity. Logical Methods in Computer Science **5**(1:6), 1–21 (2009). DOI 10.2168/LMCS-5(1:6)2009.
17. 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), 257–271 (2012). DOI 10.1007/s10472-012-9320-z.
18. Benzmüller, C., Genovese, V.: Quantified conditional logics are fragments of HOL. In: The International Conference on Non-classical Modal and Predicate Logics (NCMPL). Guangzhou (Canton), China (2011). The conference had no published proceedings; the paper is available as URL `http://arxiv.org/abs/1204.5920v1`.
19. Benzmüller, C., Paulson, L.: Exploring properties of normal multimodal logics in simple type theory with LEO-II. In: C. Benzmüller, C. Brown, J. Siekmann, R. Statman (eds.) Reasoning in Simple Type Theory — Festschrift in Honor of Peter B. Andrews on His 70th Birthday, Studies in Logic, Mathematical Logic and Foundations, pp. 386–406. College Publications (2008).
20. Benzmüller, C., Miller, D.: Automation of Higher-Order Logic. Chapter in: Gabbay, D.M., Siekmann, J.H., Woods, J. (eds.) Handbook of the History of Logic, Volume 9 — Computational Logic, North Holland, Elsevier, pp. 215-254 (2014).
21. Benzmüller, C., Paulson, L.: Multimodal and intuitionistic logics in simple type theory. The Logic Journal of the IGPL **18**(6), 881–892 (2010). DOI 10.1093/jigpal/jzp080.
22. Benzmüller, C., Paulson, L.: Quantified multimodal logics in simple type theory. Logica Universalis (Special Issue on Multimodal Logics) **7**(1), 7–20 (2013). DOI 10.1007/s11787-012-0052-y.
23. Benzmüller, C., Paulson, L.C., Sultana, N., Theiß, F.: The higher-order prover LEO-II. Journal of Automated Reasoning (2015). DOI 10.1007/s10817-015-9348-y.

24. Benzmüller, C., Sultana, N.: LEO-II version 1.5. In: J.C. Blanchette, J. Urban (eds.) PxTP 2013, *EPiC Series*, vol. 14, pp. 2–10. EasyChair (2013).
25. Benzmüller, C., Woltzenlogel Paleo, B.: Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In: T. Schaub, G. Friedrich, B. O'Sullivan (eds.) ECAI 2014, *Frontiers in Artificial Intelligence and Applications*, vol. 263, pp. 93 – 98. IOS Press (2014). DOI 10.3233/978-1-61499-419-0-93.
26. Brown, C.E.: Set comprehension in church's type theory. Ph.D. thesis, Department of Mathematical Sciences, Carnegie Mellon University (2004). See also Chad E. Brown, *Automated Reasoning in Higher-Order Logic*, College Publications, 2007
27. Brown, C.E.: Reasoning in extensional type theory with equality. In: R. Nieuwenhuis (ed.) Proc. of CADE-20, *LNCS*, vol. 3632, pp. 23–37. Springer (2005)
28. Brown, C.E.: Satallax: An automatic higher-order prover. In: B. Gramlich, D. Miller, U. Sattler (eds.) Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings, *LNCS*, vol. 7364, pp. 111–117. Springer (2012). DOI 10.1007/978-3-642-31365-3_11
29. Brown, C.E., Smolka, G.: Analytic tableaux for simple type theory and its first-order fragment. Logical Methods in Computer Science **6**(2) (2010)
30. Chellas, B.: Basic conditional logic. Journal of Philosophical Logic **4**(2), 133–153 (1975)
31. Chellas, B.: Modal Logic: An Introduction. Cambridge: Cambridge University Press (1980)
32. Church, A.: A set of postulates for the foundation of logic. Annals of Mathematics **33**(2), 346–366 (1932)
33. Church, A.: An unsolvable problem of elementary number theory. American Journal of Mathematics **58**(2), 354–363 (1936)
34. Church, A.: A formulation of the simple theory of types. Journal of Symbolic Logic **5**(2), 56–68 (1940)
35. Chwistek, L.: The Limits of Science: Outline of Logic and of the Methodology of the Exact Sciences. London: Routledge and Kegan Paul (1948)
36. Delgrande, J.: A first-order conditional logic for prototypical properties. Artif. Intell. **33**(1), 105–130 (1987)
37. Delgrande, J.: On first-order conditional logics. Artificial Intelligence **105**(1-2), 105–137 (1998)
38. Fitting, M.: Interpolation for first order S5. Journal of Symbolic Logic **67**(2), 621–634 (2002)
39. Frege, G.: Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle (1879). Translated in [43]
40. Friedman, N., Halpern, J., Koller, D.: First-order conditional logic for default reasoning revisited. ACM Transactions on Computational Logic **1**(2), 175–207 (2000)
41. Girard, J.Y.: Une extension de l'interpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In: J.E. Fenstad (ed.) 2nd Scandinavian Logic Symposium, pp. 63–92. North-Holland, Amsterdam (1971)
42. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte der Mathematischen Physik, **38**, 173–198 (1931)
43. van Heijenoort, J.: From Frege to Gödel: A Source Book in Mathematics, 1879-1931, 3rd printing, 1997 edn. Source books in the history of the sciences series. Harvard Univ. Press, Cambridge, MA (1967)
44. Henkin, L.: Completeness in the theory of types. Journal of Symbolic Logic **15**(2), 81–91 (1950)
45. Huet, G.P.: Constrained resolution: A complete method for higher order logic. Ph.D. thesis, Case Western Reserve University (1972)
46. Huet, G.P.: A mechanization of type theory. In: Proceedings of the 3rd International Joint Conference on Artificial Intelligence, pp. 139–146 (1973)
47. Kohlhase, M.: A mechanization of sorted higher-order logic based on the resolution principle. Ph.D. thesis, Saarland University (1994)
48. McDowell, R., Miller, D.: Reasoning with higher-order abstract syntax in a logical framework. ACM Transactions on Computational Logic **3**(1), 80–136 (2002)
49. Mints, G.: Cut-elimination for simple type theory with an axiom of choice. Journal of Symbolic Logic **64**(2), 479–485 (1999)

50. Muskens, R.: Intensional models for the theory of types. Journal of Symbolic Logic **72**(1), 98–118 (2007)
51. Nute, D.: Topics in conditional logic. Reidel, Dordrecht (1980)
52. Olivetti, N., Pozzato, G., Schwind, C.: A sequent calculus and a theorem prover for standard conditional logics. ACM Transactions on Computational Logic **8**(4) (2007)
53. Paleo, B.W.: An embedding of neighbourhood-based modal logics in hol. Unpublished draft; available at `https://github.com/Paradoxika/ModalLogic` (2014)
54. Pattinson, D., Schröder, L.: Generic modal cut elimination applied to conditional logics. Logical Methods in Computer Science **7**(1) (2011). DOI 10.2168/LMCS-7(1:4)2011.
55. Prawitz, D.: Hauptsatz for higher order logic. Journal of Symbolic Logic **33**(3), 452–457 (1968)
56. Ramsey, F.P.: The foundations of mathematics. In: Proceedings of the London Mathematical Society, *2*, vol. 25, pp. 338–384 (1926)
57. Rasga, J.: Sufficient conditions for cut elimination with complexity analysis. Ann. Pure Appl. Logic **149**(1-3), 81–99 (2007). DOI 10.1016/j.apal.2007.08.001.
58. Russell, B.: Mathematical logic as based on the theory of types. American Journal of Mathematics **30**(3), 222–262 (1908)
59. Schütte, K.: Semantical and syntactical properties of simple type theory. Journal of Symbolic Logic **25**(4), 305–326 (1960)
60. Smullyan, R.M.: A unifying principle for quantification theory. Proc. Nat. Acad Sciences **49**, 828–832 (1963)
61. Stalnaker, R.: A theory of conditionals. In: Studies in Logical Theory, pp. 98–112. Blackwell (1968)
62. Sutcliffe, G.: The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. Journal of Automated Reasoning **43**(4), 337-362 (2009)
63. Sutcliffe, G., Benzmüller, C.: Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. Journal of Formalized Reasoning **73**(1), 1-27 (2010)
64. Tait, W.W.: A nonconstructive proof of Gentzen's Hauptsatz for second order predicate logic. Bulletin of the American Mathematical Society **72**(6), 980983 (1966)
65. Takahashi, M.: A proof of cut-elimination theorem in simple type theory. Journal of the Mathematical Society of Japan **19**, 399–410 (1967)
66. Takeuti, G.: On a generalized logic calculus. Japanese Journal of Mathematics **23**, 39–96 (1953). Errata: ibid, vol. 24 (1954), 149–156
67. Takeuti, G.: An example on the fundamental conjecture of GLC. Journal of the Mathematical Society of Japan **12**, 238–242 (1960)
68. Takeuti, G.: Proof Theory, *Studies in Logic and the Foundations of Mathematics*, vol. 81. Elsevier (1975). DOI http://dx.doi.org/10.1016/S0049-237X(08)70603-7.
69. Wisniewski, M., Steen, A.: Embedding of First-Order Nominal Logic into HOL. In: J.-Y. Beziau et al. (eds.) Handbook of the 5th World Congress and School on Universal Logic (UNILOG'15), Istanbul, Turkey, pp. 337–339 (2015).