# Sweet *SIXTEEN*: Automation via Embedding into Classical Higher-Order Logic

Alexander Steen    Christoph Benzmüller

**Abstract**

An embedding of many-valued logics based on *SIXTEEN* in classical higher-order logic is presented. *SIXTEEN* generalizes the four-valued set of truth degrees of Dunn/Belnap's system to a lattice of sixteen truth degrees with multiple distinct ordering relations between them. The theoretical motivation is to demonstrate that many-valued logics, like other non-classical logics, can be elegantly modeled (and even combined) as fragments of classical higher-order logic. Equally relevant are the pragmatic aspects of the presented approach: interactive and automated reasoning in many-valued logics, which have broad applications in computer science, artificial intelligence, linguistics, philosophy and mathematics, become readily enabled with state of the art reasoning tools for classical higher-order logic.

**Keywords.** many-valued logic; non-classical logic; higher-order logic; automated theorem proving; semantic embedding; automation; meta-logical reasoning

## 1 Introduction

Classical logics are based on the bivalence principle, that is, the set of truth-values $V$ has cardinality $|V| = 2$, usually with $V = \{\mathrm{T}, \mathrm{F}\}$ where T and F stand for truth and falsity, respectively. Many-valued logics (MVL) generalize this requirement and allow $V$ to be a more or less arbitrary set of truth-values, often referred to as *truth-degrees*. Popular examples of many-valued logics are fuzzy logics [39, 27] with an uncountable set of truth-degrees, Gödel logics [25, 22] and Łukasiewicz logics [29] with denumerable sets of truth-degrees, and, from the class of finitely-many-valued logics, Dunn/Belnap's four-valued logic [5, 6].

The latter system, although originating from research on relevance logics, has been of strong interest to computer scientists as a formal foundation of information systems and knowledge bases. Here, the set of truth-degrees is given by the power set of $\{\mathrm{T}, \mathrm{F}\}$, i.e. $V = \{\mathbf{N}, \mathbf{T}, \mathbf{F}, \mathbf{B}\}$, where $\mathbf{N}$ denotes the empty set $\emptyset$ (mnemonic for *None*), $\mathbf{B}$ the set $\{\mathrm{T}, \mathrm{F}\}$ (for *Both*), and $\mathbf{T}$ and $\mathbf{F}$ denote the singleton sets containing the respective classical truth-value.

This article presents an approach for automating MVL based on a sixteen-valued lattice, denoted *SIXTEEN* [33]. This system has been developed by

Shramko and Wansing as a generalization of the mentioned Dunn/Belnap four-valued system to knowledge bases in computer networks [32] and was subsequently further investigated in various contexts (e.g. [31, 33]). In *SIXTEEN*, the truth-degrees are given by the power set of Dunn/Belnap's truth values, i.e. $V = 2^{\{\mathbf{N},\mathbf{T},\mathbf{F},\mathbf{B}\}}$. This generalization is essentially motivated by the observation that a four-valued system cannot express certain phenomena that arise in knowledge bases in computer networks. Further applications in linguistics and philosophy are discussed in the monograph by Shramko and Wansing [33], to which we refer for a thorough investigation.

While the use of MVL, in particular *SIXTEEN*, for knowledge representation and reasoning in computer science, linguistics and philosophy is well justified, there are unfortunately no tools available yet that support automated or interactive reasoning in *SIXTEEN*. This applies also to most other MVL systems (and the number of available systems significantly further decreases for quantified MVLs).

To that end, a semantic embedding of logics based on *SIXTEEN* within classical higher-order logic (HOL) is presented. Using this encoding, ordinary higher-order automated theorem provers can be exploited for reasoning within the many-valued setting of *SIXTEEN*. In addition, due to the expressivity of the host language, automation of meta-logical reasoning (to a certain degree) is included for free.

The semantic embedding approach provides similar results for other non-classical logics, yielding out-of-the-box automation of many other logics using ordinary HOL reasoning systems. Most recent related work has been done in the context of automation of higher-order quantified modal logic [10, 15], quantified conditional logics [8], quantified hybrid logics [35] and free logics [12]. There is empirical evidence that such tools can be employed to successfully verify or refute non-trivial arguments in e.g. metaphysics and that they even can contribute new knowledge [14, 13, 7].

The remainder of this article is organized as follows: In §2, the above mentioned logics based on *SIXTEEN* are introduced. §3 and §4 address HOL and its utilization for automating reasoning within MVL. Subsequently, in §5 experiments with the aforementioned encoding are displayed and discussed. Finally, §6 concludes the article and sketches further extensions of the presented approach.

## 2 Many-Valued Logics Based on *SIXTEEN*

The MVL systems addressed here are, as outlined earlier, based on a sixteen-valued structure of truth-degrees. The underlying set $V$ of truth-degrees is given by the power set of the power set of the classical (bivalent) truth-values $\{\mathrm{T}, \mathrm{F}\}$, i.e. $V := 2^{2^{\{\mathrm{T},\mathrm{F}\}}}$. The set $V$ thus further generalizes the set of truth-degrees of Dunn/Belnap's system. More precisely, we have

$$V = 2^{\{\mathbf{N},\mathbf{T},\mathbf{F},\mathbf{B}\}} = \{\mathbf{N}, \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{B}, \mathbf{NT}, \ldots, \mathbf{NTFB}\}$$

where $\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{F}$ and $\boldsymbol{B}$ are the respective singleton sets containing $\mathbf{N}, \mathbf{T}, \mathbf{F}$ and $\mathbf{B}$. The remaining truth-degrees are named using a combination of the letters $\mathbf{N}, \mathbf{T}, \mathbf{F}$ and $\mathbf{B}$, representing the truth-degree that contains the respective elements when regarded as a set (e.g. $\mathbf{NT}$ for the set $\{\mathbf{N}, \mathbf{T}\}$).

Using the above set $V$, there are multiple, mutually independent, possibilities on how to order the truth-degrees in a meaningful way. They can, for instance, be sorted by increasing truth. But there are other reasonable orderings one can think of, e.g. when interested in the decrease of falsity (which is not the same thing as increase of truth).

Shramko and Wansing [33, pp. 53–57] suggest three reasonable independent (partial) orderings, for the set of truth-degrees $V$.

First, the ordering $\leq_i$ orders elements of $V$ by *information*. Here, a truth-degree $v$ is smaller than $w$ with respect to its information value, if and only if $v$ is a subset of $w$, i.e. $v \leq_i w :\Leftrightarrow v \subseteq w$. The remaining two orderings which are more suited for logical reasoning are $\leq_t$ and $\leq_f$, comparing truth-degrees by their *truth* and *falsity*, respectively. For a formal definition of these orderings, the notions of "truthful" and "truthless" subsets of a truth-degree $v$ are introduced: The truthful subset of $v$, denoted $v^t$, contains exactly those elements in $v$ which themselves contain T. The truthless subset of $v$, denoted $v^{-t}$, accordingly consists only of those elements of $v$ which do not contain T. This notion is analogously extended to $(.)^f$ and $(.)^{-f}$. More formally, we have

$$\begin{aligned} v^t &:= \{x \in v \mid \mathrm{T} \in x\} \\ v^{-t} &:= \{x \in v \mid \mathrm{T} \notin x\} \end{aligned} \tag{1}$$

and for its counterpart based on falsity

$$\begin{aligned} v^f &:= \{x \in v \mid \mathrm{F} \in x\} \\ v^{-f} &:= \{x \in v \mid \mathrm{F} \notin x\} \end{aligned} \tag{2}$$

Note that $v^t \neq v^{-f}$ and $v^f \neq v^{-t}$, i.e. the two respective counterparts of these sets do not coincide. As it is pointed out by Shramko and Wansing, the counterparts of these notions do indeed coincide for the four-valued system of Dunn/Belnap [33, p.53]. That is why there is a single unique *logical ordering* in that system, as opposed to the system described here.

The ordering $\leq_t$ can elegantly be defined as an increase in truth and a non-increase of non-truth. Analogously, $\leq_f$ orders by increase of falsity and non-increase of non-falsity:

$$\begin{aligned} v \leq_t w &:\Leftrightarrow v^t \subseteq w^t \wedge w^{-t} \subseteq v^{-t} \\ v \leq_f w &:\Leftrightarrow v^f \subseteq w^f \wedge w^{-f} \subseteq v^{-f} \end{aligned} \tag{3}$$

The above orderings $\leq_i, \leq_t$ and $\leq_f$ induce a so-called *trilattice* [34]

$$\mathit{SIXTEEN} = (V, \sqcup_i, \sqcap_i, \sqcup_t, \sqcap_t, \sqcup_f, \sqcap_f)$$

which is essentially a threefold lattice, i.e. having three mutually independent pairs of meet and join operations.

Additionally to the above meet and join operations, there are inversion operations, denoted by $-_\Box$, for each axis $\Box \in \{t, f, i\}$ of the trilattice. These inversions generalize the notion of *conflation* [23] to trilattices [34]. As for conflation, the key property of a specific inversion operation is that it inverses only one ordering while not changing the order with respect to the other axes. For instance, if $v \leq_t w$, then $(-_t\ w) \leq_t (-_t\ v)$, but still $(-_f\ v) \leq_t (-_f\ w)$ and $(-_i\ v) \leq_t (-_i\ w)$, i.e. ordering by truth is invariant under $f$-inversion and $i$-inversion. More formally, for $\Box, \diamond \in \{t, f, i\}$, $\Box \neq \diamond$, an operation $-_\Box$ is an inversion with respect to axis $\Box$ if $-_\Box$ has the following properties [33, p. 58]:

$$\begin{cases} v \leq_\Box w & \Rightarrow -_\Box w \leq_\Box -_\Box v \\ v \leq_\diamond w & \Rightarrow -_\Box v \leq_\diamond -_\Box w \\ -_\Box -_\Box v & = v \end{cases} \tag{4}$$

We are now sufficiently prepared to present the syntax and semantics for the respective logics based on truth- and falsity-orderings. The three logics studied in the remainder are denoted as $\mathcal{L}_t$, $\mathcal{L}_f$, and $\mathcal{L}_{tf}$. Their abstract syntax is given as:

$$\begin{aligned} \mathcal{L}_t :\ & A, B ::= x \mid A \wedge_t B \mid A \vee_t B \mid\ \sim_t A \\ \mathcal{L}_f :\ & A, B ::= x \mid A \wedge_f B \mid A \vee_f B \mid\ \sim_f A \\ \mathcal{L}_{tf} :\ & A, B ::= x \mid A \wedge_t B \mid A \vee_t B \mid\ \sim_t A \mid A \wedge_f B \mid A \vee_f B \mid\ \sim_f A \end{aligned}$$

where $x$ is a propositional variable, and $\wedge$, $\vee$, and $\sim$ are the respective connectives for conjunction, disjunction and negation. The primary focus is on $\mathcal{L}_{tf}$ since the other languages are proper fragments of it.

To provide a semantics for the above languages, let $v^{16}$ be a *16-valuation*, that is, a map from propositional variables to the sixteen-valued set $V$. The semantic evaluation of propositional variables is extended to compound formulae as usual ($\Box \in \{t, f\}$):

$$\begin{aligned} v^{16}(A \wedge_\Box B) &:= v(A) \sqcap_\Box v(B) \\ v^{16}(A \vee_\Box B) &:= v(A) \sqcup_\Box v(B) \\ v^{16}(\sim_\Box A) &:= -_\Box v(A) \end{aligned} \tag{5}$$

Semantic entailment can now be defined as an increase in truth or as an decrease in falsity. More formally, for two arbitrary formulas $A, B \in \mathcal{L}_{tf}$, $A$ entails $B$ wrt. to truth order, $A \models^{16}_t B$, if and only if $v^{16}(A) \leq_t v^{16}(B)$ for all 16-valuations $v^{16}$. Analogously we have $A \models^{16}_f B$ if and only if $v^{16}(B) \leq_f v^{16}(A)$, for all 16-valuations $v^{16}$. The resulting logics are $(\mathcal{L}_{tf}, \models^{16}_t)$, $(\mathcal{L}_{tf}, \models^{16}_f)$ and the *bi-consequence logic* $(\mathcal{L}_{tf}, \models^{16}_t, \models^{16}_f)$ [33, p. 65].

# 3 Classical Higher-Order Logic

Higher-order logic (HOL) is an elegant and expressive formal system that extends first-order logic with quantification over arbitrary sets and functions.

Church [21] proposed a version of higher-order logic, called simple type theory (in the following referred to as HOL), which he built on top of the simply typed $\lambda$-calculus [19, 20]. The simply typed $\lambda$-calculus augments the untyped $\lambda$-calculus, as studied by Alonzo Church in the 1930s, with *simple types*. The set of simple types $\mathcal{T}$ is thereby freely generated from a set of base types and a function type constructor. In HOL, the set of base types is usually taken as (a superset of) $\{\iota, o\}$ with $\iota$ and $o$ being the type of individuals and classical truth values, respectively.

**Syntax of HOL.** The terms of the logic are essentially those of the simply typed $\lambda$-calculus, enriched with typed (logical) constants. These constants are taken from a family of denumerable sets of constant symbols $\Sigma := (\Sigma_\tau)_{\tau \in \mathcal{T}}$, called *signature*. Together with a family of typed variable symbols $(\mathcal{V}_\tau)_{\tau \in \mathcal{T}}$ the terms of HOL are then those terms contained in the smallest set $\Lambda$ for which the following conditions hold: Each constant symbol $c_\tau \in \Sigma_\tau$ and each variable symbol $X_\tau \in \mathcal{V}_\tau$ is a HOL term of type $\tau$. If $X_\tau \in \mathcal{V}_\tau$ is a variable symbol and $s_\nu \in \Lambda$ is a HOL term, then the *abstraction* (of $s_\nu$) $(\lambda X_\tau.\ s_\nu)_{\nu\tau} \in \Lambda$ is a HOL term of type $\nu\tau$. Finally, if $s_\tau, t_{\nu\tau} \in \Lambda$ are HOL terms, then the *application* (of $t_{\nu\tau}$ onto $s_\tau$) given by $(t_{\nu\tau}\ s_\tau)_\nu \in \Lambda$ is a HOL term of type $\nu$. Hereby $\tau, \nu \in \mathcal{T}$ are types and the *abstraction type* $\nu\tau$ denotes the type of functions from arguments of type $\tau$ to values of type $\nu$. Abstraction types are considered left-associative, i.e. $\tau\nu\mu \equiv (\tau\nu)\mu$. As usual for Church-style typing, a term's type is given as subscript and considered a part of its name, hence intrinsic to it. Nevertheless, type subscripts are omitted in the following if clear from the context.

We choose the signature $\Sigma$ to consist at least of the *primitive logical connectives*, that are negation $\neg_{oo}$, disjunction $\vee_{ooo}$, and universal quantification $\Pi^\tau_{o(o\tau)}$ for each type $\tau \in \mathcal{T}$. The remaining (non-primitive) logical connectives can be defined as abbreviations in the usual way. By T and F we denote the HOL symbols for truth and falsehood, respectively. T can e.g. be defined as $\mathrm{T} := \Pi^o(\lambda X_o.\ X \vee \neg X)$.

We use *binder notation* $\forall X_\tau.\ s_o$ as shorthand for universal quantification given by $\Pi^\tau_{(\tau \to o) \to o}(\lambda X_\tau.\ s_o)$. For additional convenience, we allow infix notation for the common binary logical connectives, i.e. write $(s \vee t)$ instead of $((\vee\ s)\ t)$.

A *formula* of HOL is a term $s_o \in \Lambda$, hence of type $o$. As usual, a *sentence* is a closed formula.

**Semantics of HOL.** The usual rules of $\lambda$-conversions ($\alpha$-, $\beta$-, and $\eta$-conversion) are intrinsically included in HOL. Using these conversions, especially $\beta$-reduction, all quantifier instantiations can be expressed very concisely.

The meta-operation of *substituting* a variable $X_\tau$ by some term $t_\tau$ in $s_\nu$ is denoted $s[X_\tau/t_\tau]$. Hereby, we assert that there is no variable capture happening in $s_\tau$ by assuming $\alpha$-conversion as implicit when necessary. A $\beta$-redex of the form $(\lambda X_\tau.\ s_\nu)\ t_\tau$ then $\beta$-reduces to $s_\nu[X_\tau/t_\tau]$. A term $s_\tau$ is said to be in $\beta$-normal form if it does not contain any $\beta$-redex as a subterm. The $\beta$-normal form of $s_\tau$ is denoted $s_\tau{\downarrow_\beta}$, equivalence modulo $\beta$-conversion (and $\alpha$-conversion) is denoted $=_\beta$. Reduction, normal forms and equivalence modulo $\eta$ and $\beta\eta$ are defined analogously. We refer to the literature for a thorough study of typed $\lambda$-calculi [4].

The semantics of HOL is meanwhile well-understood [9] and various semantic generalizations have been studied: We here summarize the most important points: As a consequence of Gödel's incompleteness theorem [24], the so-called *standard semantics* of HOL is necessarily incomplete. However, it shows that in many practical applications Henkin's weaker form of *general semantics* [28, 2, 1] is sufficiently expressive. For Henkin's generalized semantics sound and complete proof calculi exists. And such proof calculi provide the theoretical foundations of modern theorem provers for HOL such as LEO-II [11], Leo-III [36] and Satallax [18].

Next, standard and Henkin semantics are introduced more formally. We start out with the notion of frames.

A *frame* is a collection $\{\mathcal{D}_\tau\}_{\tau\in\mathcal{T}}$ of non-empty sets (called *domains*), where $\mathcal{D}_o$ is the domain of classical truth-values, chosen to be $\mathcal{D}_o = \{T, F\}$ (for truth and falsehood, respectively) and sets $\mathcal{D}_{\nu\tau}$, which denote the domain of functions of type $\nu\tau$ and range over functions from domain $\mathcal{D}_\tau$ to co-domain $\mathcal{D}_\nu$. The domain of individuals $\mathcal{D}_\iota$ is not further restricted (except for being non-empty).

A *HOL model* $M$ is a pair $M = (\{\mathcal{D}_\tau\}_{\tau\in\mathcal{T}}, I)$, where $\{\mathcal{D}_\tau\}_{\tau\in\mathcal{T}}$ is a frame and $I$ is a function that maps each constant symbol $c_\tau \in \Sigma_\tau$ to an element of $\mathcal{D}_\tau$ (the *denotation of* $c_\tau$). It is assumed that $I$ is chosen such that the logical connectives $\neg_{oo}$, $\vee_{ooo}$ and $\Pi_{o(o\alpha)}$ have their usual meaning, e.g. $I(\neg_{oo})$ is the set-theoretic function that inverts the truth-values of $\mathcal{D}_o$:

$$I(\neg_{oo}) = v \mapsto \begin{cases} T & \text{if } v = F \\ F & \text{if } v = T \end{cases}$$

A variable assignment $g$ is a map that assigns each variable $X_\tau \in \mathcal{V}_\tau$ an element in $\mathcal{D}_\tau$. With $g[Y/s]$ we mean the variable assignment that is identical to $g$ except that variable $Y$ is now mapped to $s$.

Finally, given a HOL model $M = (\{\mathcal{D}_\tau\}_{\tau\in\mathcal{T}}, I)$ and a variable assignment $g$, the *value* of a HOL term (with respect to $M$ and $g$), denoted by $\|.\|^{M,g}$, is given by

(i) $\|X_\tau\|^{M,g} = g(X_\tau)$ and $\|c_\tau\|^{M,g} = I(c_\tau)$

(ii) $\|(s_{\nu\tau}\ t_\tau)\|^{M,g} = \|s_{\nu\tau}\|^{M,g}\ \|t_\tau\|^{M,g}$

(iii) $\|(\lambda X_\tau.\ s_\nu)\|^{M,g}$ is a function $f \in \mathcal{D}_{\nu\tau}$ s.t. for all $z \in \mathcal{D}_\tau$ it holds that $f(z) = \|s_\nu\|^{M,g[X_\tau/z]}$

A HOL model $M$ is called a *standard model* if and only if the sets $\mathcal{D}_{\nu\tau}$ are chosen to be the complete set $\mathcal{D}_\nu^{\mathcal{D}_\tau}$ of functions from domain $\tau$ to co-domain $\nu$. The notion of general models (or *Henkin models*) is, in contrast, defined by choosing $\mathcal{D}_{\nu\tau}$ as a subset of $\mathcal{D}_\nu^{\mathcal{D}_\tau}$ such that it contains "sufficiently many", but not necessarily all, functions. More formally, $M$ is a general model if and only if $\|.\|^{M,g}$ is a total function (that is, every term is assigned a value). The function $\|.\|^{M,g}$ is uniquely determined for every general model. Of course, every standard model is also a general model.

For a model $M$ and a variable assignment $g$, a formula $s_o$ is *valid in $M$ wrt. variable assignment $g$*, denoted $M, g \models^{HOL} s_o$, if $\|s_o\|^{M,g} = T$. It is called *valid in $M$* if $M, g \models^{HOL} s_o$ for all variable assignments $g$. This is written as $M \models^{HOL} s_o$. Finally, a formula $s_o$ is called *Henkin-valid* (or simply *valid*), written $\models^{HOL} s_o$, if $s_o$ is valid in every Henkin model. In the following, we always assume general semantics of HOL (i.e. only Henkin models).

# 4 Embedding of $\mathcal{L}_{tf}$

This section presents a semantic embedding of the logic $\mathcal{L}_{tf}$ – and thereby automatically also of $\mathcal{L}_t$ and $\mathcal{L}_f$ – in HOL. The idea is essentially to exploit the expressiveness of HOL for encoding the semantics of the given truth-degrees and the operations on them.

The truth-degrees of *SIXTEEN* have been introduced as sets of sets of (classical) truth-values T and F (cf. §2). Note that sets can be elegantly represented via characteristic functions in HOL and $\lambda$-abstraction can be utilized for this purpose. Exploiting this idea, we below encode all sixteen sets that match the respective truth-degree. More precisely, a set $M = \{x \mid P(x)\}$ is modeled in HOL by its *characteristic function* $\chi_M = \lambda x.P\ x$, which is a predicate that holds for any element $m$ contained in $M$ and does not hold for any other element $m \notin M$. The $\lambda$-abstractions of HOL are typed. For example, the characteristic function for a set of truth values, $\lambda x_o.P\ x$, has type *oo*. Consequently, the characteristic function for a set of sets of truth values has type $o(oo)$. Thus, truth-degrees of *SIXTEEN* correspond to functions of type $o(oo)$. A single truth-degree is then a function $\lambda n_{oo}.\ P\ n$ where $P_{o(oo)}$ is an explicit predicate that – via function application – determines which elements are to be contained within (the set) $n$ so that $n$ is itself contained in the set (of sets) under consideration. In other words the sets as characteristic functions approach is applied here in nested fashion.

The usual set operations are defined as follows (we use the conventional infix notation):

$$s_{o\tau} \cup t_{o\tau} := \lambda x_\tau.\ (s\ x) \vee (t\ x)$$
$$s_{o\tau} \cap t_{o\tau} := \lambda x_\tau.\ (s\ x) \wedge (t\ x)$$

where $\tau \in \mathcal{T}$ is some type.

As an example, consider the truth-degree $\boldsymbol{N}$, which corresponds to the set $\{\emptyset\}$, i.e. the set only containing the empty set of truth values. Note that this set $\boldsymbol{N}$ contains exactly those sets of truth values that neither contain T nor F; the empty set of truth values is hence the sole candidate fulfilling this condition. Consequently, our encoding of $\boldsymbol{N}$ is $\lambda n_{oo}.\ \neg(n\ \mathrm{F}) \wedge \neg(n\ \mathrm{T})$. The list of all sixteen truth degrees and their respective encoding in HOL is presented at Table 1.

| | | |
|---|---|---|
| **N** | = | $\lambda n_{oo}.\ \mathrm{F}$ |
| ***N*** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{F}) \wedge \neg(n\ \mathrm{T})$ |
| ***T*** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{F}) \wedge n\ \mathrm{T}$ |
| ***F*** | = | $\lambda n_{oo}.\ n\ \mathrm{F} \wedge \neg(n\ \mathrm{T})$ |
| ***B*** | = | $\lambda n_{oo}.\ n\ \mathrm{F} \wedge n\ \mathrm{T}$ |
| **NF** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{T})$ |
| **NT** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{F})$ |
| **NB** | = | $\lambda n_{oo}.\ (\neg(n\ \mathrm{F}) \wedge \neg(n\ \mathrm{T})) \vee (n\ \mathrm{F} \wedge n\ \mathrm{T})$ |
| **FT** | = | $\lambda n_{oo}.\ (n\ \mathrm{F} \wedge \neg(n\ \mathrm{T})) \vee (\neg(n\ \mathrm{F}) \wedge n\ \mathrm{T})$ |
| **FB** | = | $\lambda n_{oo}.\ n\ \mathrm{F}$ |
| **TB** | = | $\lambda n_{oo}.\ n\ \mathrm{T}$ |
| **NFT** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{F}) \vee \neg(n\ \mathrm{T})$ |
| **NFB** | = | $\lambda n_{oo}.\ n\ \mathrm{F} \vee \neg(n\ \mathrm{T})$ |
| **NTB** | = | $\lambda n_{oo}.\ \neg(n\ \mathrm{F}) \vee n\ \mathrm{T}$ |
| **FTB** | = | $\lambda n_{oo}.\ n\ \mathrm{F} \vee n\ \mathrm{T}$ |
| **NFTB** | = | $\lambda n_{oo}.\ \mathrm{T}$ |

Table 1: Encoding of all truth-degrees of *SIXTEEN* in HOL.

The appropriateness of the encodings can be shown by verifying that the set-theoretic denotation of the characteristic function representing the truth-degree is indeed isomorphic (denoted $\simeq$) to its interpretation as (set-theoretic) set.

**Lemma 4.1.** *Let $\lceil v \rceil$ denote the HOL encoding of truth-degree $v \in V$ (as given in Table 1) and let $v \in V$ be a truth-degree. Then, for any HOL model $M$ and variable assignment $g$ it holds that $\| \lceil v \rceil \|^{M,g} \simeq v$.*

*Proof.* Simple application of definitions and the fact that $\|\mathrm{T}\|^{M,g} = T$ and $\|\mathrm{F}\|^{M,g} = F$ for all HOL models $M$ and variable assignments $g$. □

We now present the encoding of the logical operations of $\mathcal{L}_{tf}$. Recall that their semantics is defined using the lattice operations $\sqcup, \sqcap$ and the inversion operation $-$ as introduced in §2.

Again, the notion of truthful subsets $(.)^t$ and truthless subsets $(.)^{-t}$ (cf. Eq. (1)) is needed to define the ordering $\leq_t$ on truth-degrees:

$$(v)^t_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge (n\ \mathrm{T}) \qquad (v)^{-t}_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge \neg(n\ \mathrm{T})$$

For any truth degree $v$, $(v)^t$ is itself again a set of sets of truth-values, hence its encoding is similar to that of truth-degrees. Here, the sub-expression $(v\ n)$ asserts that $n$ is contained in $v$, and $(n\ \mathrm{T})$ ensures that T is contained in $n$. The analogous embedding of Eq. (2) is given by

$$(v)^f_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge (n\ \mathrm{F}) \qquad (v)^{-f}_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge \neg(n\ \mathrm{F})$$

The orderings $\leq_t$ and $\leq_f$ can then be encoded to match the definition of Eq. (3) using the same techniques as before:

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ \big((v^t\ n) \Rightarrow (w^t\ n)\big) \wedge \big((w^{-t}\ n) \Rightarrow (v^{-t}\ n)\big)$$
$$\leq_f := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ \big((v^f\ n) \Rightarrow (w^f\ n)\big) \wedge \big((w^{-f}\ n) \Rightarrow (v^{-f}\ n)\big)$$

**Lemma 4.2.** *Let $\Box \in \{t, f\}$ and let $\lceil \leq_\Box \rceil$ denote the HOL encoding of $\leq_\Box$. Then, for $v, w \in V$, it holds that $\models^{HOL} \big(\lceil \leq_\Box \rceil\ \lceil v \rceil\ \lceil w \rceil\big)$ if and only if $v \leq_\Box w$.*

*Proof.* By Lemma 4.1, we know that $\lceil v \rceil \simeq v$ for any $v \in V$. The remaining embeddings are well-known set operations of HOL. $\Box$

The embedding of $\sqcup, \sqcap$ and $-$ are slightly more complicated as we need a closed algebraic description for these operators. In the original description, only an implicit characterization via properties is given for each of these operations. Up to the authors' knowledge, there has not been any such closed algebraic formulation in the literature. As it turns out, the join and meet operations can be defined as

$$\sqcup_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^t \cup w^t \cup \big(w^{-t} \cap v^{-t}\big)$$
$$\sqcap_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^{-t} \cup w^{-t} \cup \big(w^t \cap v^t\big)$$
$$\sqcup_f := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^f \cup w^f \cup \big(w^{-f} \cap v^{-f}\big)$$
$$\sqcap_f := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^{-f} \cup w^{-f} \cup \big(w^f \cap v^f\big)$$

The intuition behind these definitions is as follows: Join operations (here for $\sqcup_t$) construct a set that combines the "truthful" elements of the truth-degree while only containing those "truthless" elements that were contained in both sets. Note that this is compatible with the ordering idea of $\leq_t$, where bigger elements increase $(.)^t$ but do not increase $(.)^{-t}$. A similar argumentation holds for the meet operations, yielding smaller elements with respect to the respective ordering.

**Lemma 4.3.** *Let $\Box \in \{t, f\}$ and $\lceil \sqcup_\Box \rceil$ denote the HOL encoding of $\sqcup_\Box$. Then, for $v, w \in V$ and every HOL model $M$ and variable assignment $g$, it holds that $\|\lceil \sqcup_\Box \rceil\ \lceil v \rceil\ \lceil w \rceil\|^{M,g} \simeq v \sqcup_\Box w$.*

*Proof.* By Lemma 4.1 and 4.2, we know that $\lceil v \rceil$ and $\lceil \leq_\Box \rceil$ are appropriate embeddings, for $v \in V$, $\Box \in \{t, f\}$. Since joins are unique, if they exist, it suffices to show that $\|\lceil \sqcup_\Box \rceil\ \lceil v \rceil\ \lceil w \rceil\|^{M,g}$ is indeed a join of $\lceil v \rceil$ and $\lceil w \rceil$. $\Box$

**Lemma 4.4.** *Let $\Box \in \{t, f\}$ and $\lceil \sqcap_\Box \rceil$ denote the HOL encoding of $\sqcap_\Box$. Then, for $v, w \in V$ and every HOL model $M$ and variable assignment $g$, it holds that $\|\lceil \sqcap_\Box \rceil\ \lceil v \rceil\ \lceil w \rceil\|^{M,g} \simeq v \sqcap_\Box w$.*

*Proof.* Analogous to the proof of Lemma 4.3 $\Box$

Finally, the inversion operation $-_t v$ can be encoded by explicitly constructing sets $(\lambda b_o. \cdots)$ for each element $n$ of the original truth-degree $v$ such that it contains T whenever $n$ does not contain T, and it contains F if and only if F is contained in $n$. That way we only swap the property whether an element of $v$ contains T, hence inverting it with respect to $\leq_t$:

$$-_t := \lambda v_{o(oo)}.\lambda n_{oo}.\, v\ \big(\lambda b_o.(\neg b \Rightarrow n\ \mathrm{F}) \wedge (b \Rightarrow \neg(n\ \mathrm{T}))\big)$$

An analogous construction is employed for $-_f v$, where elements of $v$ containing F are swapped for elements that do not contain F but still contain T if they originally did:

$$-_f := \lambda v_{o(oo)}.\lambda n_{oo}.\, v\ \big(\lambda b_o.(\neg b \Rightarrow \neg(n\ \mathrm{F})) \wedge (b \Rightarrow n\ \mathrm{T})\big)$$

**Lemma 4.5.** *Let $\Box \in \{t, f\}$ and $\lceil -_\Box \rceil$ denote the HOL encoding of $-_\Box$. Then, for $v \in V$ and every HOL model $M$ and variable assignment $g$, it holds that $\| \lceil -_\Box \rceil\ \lceil v \rceil \|^{M,g} \simeq -_\Box v$.*

*Proof.* Since both $-_t$ and $-_f$ are uniquely determined [33, Table 3.1, p. 58], it suffices to verify each of the sixteen cases for both operations. Simple calculation confirms that each $v \in V$ is mapped to the appropriate inverse $-_\Box v$. $\Box$

All three entailment relations $\models^{16}_t, \models^{16}_f$ and $\models^{16}_{tf}$ can be expressed by the above definitions since they are defined via increase of truth (or decrease of falsity), i.e. by means of $\leq_t$ and $\leq_f$.

**Soundness and Completeness** Using the afore stated results, we can now prove soundness and completeness of the embedding of $\mathcal{L}_{tf}$.

**Theorem 4.6.** *Let $\Phi, \Psi$ be $\mathcal{L}_{tf}$ formulas and let $\lceil \Phi \rceil, \lceil \Psi \rceil$ be the corresponding embedded formulas in HOL according to our encoding from above. It holds that*

$$\Phi \models^{16}_t \Psi \quad \textit{iff} \quad \models^{HOL} \lceil \leq_t \rceil\ \lceil \Phi \rceil\ \lceil \Psi \rceil$$

*and*

$$\Phi \models^{16}_f \Psi \quad \textit{iff} \quad \models^{HOL} \lceil \leq_f \rceil\ \lceil \Psi \rceil\ \lceil \Phi \rceil$$

*Proof.* Simple application of the above lemmas. $\Box$

**Corollary 4.7.** *Let $\Phi, \Psi$ be $\mathcal{L}_{tf}$ formulas and let $\lceil \Phi \rceil, \lceil \Psi \rceil$ be the corresponding embedded formulas in HOL according to our encoding from above. It holds that*

$$\Phi \models^{16}_{tf} \Psi \quad \textit{iff} \quad \models^{HOL} \big(\lceil \leq_t \rceil\ \lceil \Phi \rceil\ \lceil \Psi \rceil\big) \vee \big(\lceil \leq_f \rceil\ \lceil \Psi \rceil\ \lceil \Phi \rceil\big)$$

Theorem 4.6 and Corollary 4.7 now enable us to employ standard HOL reasoning systems for reasoning within $\mathcal{L}_{tf}$ (and its sublanguages $\mathcal{L}_t$ and $\mathcal{L}_f$). Since these systems can access the meta-logical definitions of *SIXTEEN* operations during proof search, we also can automatically prove (to some degree) meta-logical results of these logics. In fact, all of the key properties of $\leq_\Box, \sqcup_\Box, \sqcap_\Box, -_\Box$ given by Shramko and Wansing in their monograph [33, Chapter 3.5] have been verified automatically for the presented embedding.

## 5 Experiments and Results

To enable experiments and further utilization of the embedding in practice, we have encoded the above embedding in TPTP THF syntax [37, 38], which is a concrete syntax format for HOL. An excerpt of this TPTP THF encoding is given in Fig. 1. Altogether this encoding consists of approx. 150 lines of code, including comments. It can simply be loaded as axiomatization file by any TPTP-compatible HOL ATP for reasoning within $\mathcal{L}_{tf}$. Additionally, we provided the embedding as a theory for the renowned interactive proof assistant Isabelle/HOL [30]. As a proof of concept for the practical usability of our automation approach, we formulated several proof tasks within and about *SIXTEEN* for ATP systems.

Concerning object-level reasoning within $\mathcal{L}_{tf}$, we formulated small exemplary proof problems. The most interesting problems for this kind of reasoning might be those which employ a joint "truth-falsehood framework", i.e. where the proof problem contains mixed truth- and falsehood-based operators. As an example, consider the valid entailment $A \wedge_t B \models_t A$. If we now, however, assume entailment by decrease of falsity, neither $A$ nor $B$ can be inferred from $A \wedge_t B$. More formally,

$$A \wedge_t B \not\models_f A \quad \text{and} \quad A \wedge_t B \not\models_f B$$

Proving the first entailment and disproving the two latter entailments is an easy task for the employed HOL provers that use our embedding. In fact, also counter-model finders for HOL, such as Nitpick [17], can be used in order to find a concrete counter example to the last two invalid entailments. Table 2 shows the object-level reasoning benchmarks.

| **Statement** | **Result** | **Time** |
|---|---|---|
| $A \wedge_t B \models_t A$ | Theorem | $5ms$ |
| $A \wedge_t B \models_f A$ | Countersatisfiable | $6ms$ |
| $\sim_f (A \wedge_t B) \models_t \sim_f A$ | Theorem | $8ms$ |
| $\sim_f \sim_t A \models_t \sim_t \sim_f A$ | Theorem | $5ms$ |
| $A \wedge_f B \models_t A \vee_t B$ | Theorem | $6ms$ |
| $A \wedge_t B \models_f A \vee_t B$ | Countersatisfiable | $7ms$ |

Table 2: Automated verification results of object-level reasoning tasks. The time results refer to the measurements with Satallax 2.7.

Another interesting suite of experiments aims at verifying the correctness of our closed formulations and encoding of the lattice operations $\sqcup, \sqcap$ and $-$. To that end, we have checked its definitions against the respective properties given in the monograph of Shramko and Wansing [33, Prop. 3.2, Def. 3.6]. Table 3 displays the respective properties that have been given to ordinary HOL theorem provers. The automatically verified meta-logical encodings empirically confirm that our embedding indeed captures the intended semantics.

```
%-- Truth degrees
thf(n_type,type,( n: ($o>$o)>$o )).
thf(n_def,definition,( n=(^[X:$o>$o]:$false) )).
thf(nn_type,type,( nn: ($o>$o)>$o )).
thf(nn_def,definition,( nn = (^[X:$o>$o]:(~(X@$false)&~(X@$true))) )).
...
thf(ftb_type,type,( ftb: ($o>$o)>$o )).
thf(ftb_def,definition,(ftb = (^[X:$o>$o]:((X@$false)|(X@$true))) )).
thf(all_type,type,( all: ($o>$o)>$o )).
thf(all_def,definition,( all = (^[X:$o>$o]:$true) )).
%-- Truthful/Truthless subsets
thf(tpos_subset_type,type,( tpos_subset: ((($o>$o)>$o)>($o>$o)>$o) )).
thf(tpos_subset_def,definition,( tpos_subset =
 (^[T:($o>$o)>$o,X:$o>$o]:((T@X)&(X@$true))) )).
thf(tneg_subset_type,type,( tneg_subset: ((($o>$o)>$o)>($o>$o)>$o) )).
thf(tneg_subset_def,definition,( tneg_subset =
 (^[T:($o>$o)>$o,X:$o>$o]:((T@X)&~(X@$true))) )).
...
%-- Orderings
thf(ord_t_type,type,( ord_t: ((($o>$o)>$o)>(($o>$o)>$o)>$o) )).
thf(ord_t_def,definition,( ord_t = (^[X:($o>$o)>$o,Y:($o>$o)>$o]:
 (![A:$o>$o]: (((tpos_subset@X@A)=>(tpos_subset@Y@A))
             &((tneg_subset@Y@A)=>(tneg_subset@X@A))))) )).
...
%-- Lattice operations
thf(inverse_t_type,type,( inverse_t: ((($o>$o)>$o)>($o>$o)>$o) )).
thf(inverse_t_def,definition,( inverse_t = (^[T:($o>$o)>$o,X:$o>$o]:
 (T@(^[Y:$o]:((~(Y)=>(X@$false))&(Y=>~(X@$true)))))) )).
thf(join_t_ty,type,( join_t: (($o>$o)>$o)>(($o>$o)>$o)>($o>$o)>$o )).
thf(join_t_def,definition,( join_t = (^[X:($o>$o)>$o,Y:($o>$o)>$o]:
 (union@(union@(tpos_subset@X)@(tpos_subset@Y))
       @(intersect@(tneg_subset@X)@(tneg_subset@Y)))) )).
thf(meet_t_ty,type,( meet_t: (($o>$o)>$o)>(($o>$o)>$o)>($o>$o)>$o )).
thf(meet_t_def,definition,( meet_t = (^[X:($o>$o)>$o,Y:($o>$o)>$o]:
(union@(union@(tneg_subset@X)@(tneg_subset@Y))
      @(intersect@(tpos_subset@X)@(tpos_subset@Y)))) )).
...
```

Figure 1: THF encoding excerpt. Some truth-degrees and operations are omitted for brevity. Some notes concerning the THF format: The type of truth-values is written `$o`, and `$o>$o` represents the type of (characteristic) functions from truth-values to truth-values, etc. `$true` and `$false` represent truth and falsity. $\lambda$-abstractions and applications are denoted with `^` and `@`, respectively. `~`, `|`, `&`, `=>` encode negation, disjunction, conjunction and implication, and `!` denotes universal quantification. Comments are lines starting with `%`.

For our measurements, the two automated theorem provers LEO-II [11] and Satallax [18] were used. As it can be seen, all proof tasks were solved successfully, which provides strong evidence in addition to theoretical results above, for the soundness (and completeness) of our embedding. In most cases, the desired properties could be automatically proved in less than $10ms$. This provides further evidence for the practical relevance of our approach.

| **Source** | **Statement** | **Result** | **Time** |
|---|---|---|---|
| Prop 3.2 1. | $\forall s,t.(\mathbf{T} \in s \wedge \mathbf{T} \in t) \Leftrightarrow \mathbf{T} \in s \sqcap_t t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{B} \in s \wedge \mathbf{B} \in t) \Leftrightarrow \mathbf{B} \in s \sqcap_t t$ | Theorem | $9ms$ |
| | $\forall s,t.(\mathbf{F} \in s \vee \mathbf{F} \in t) \Leftrightarrow \mathbf{F} \in s \sqcap_t t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{N} \in s \vee \mathbf{N} \in t) \Leftrightarrow \mathbf{N} \in s \sqcap_t t$ | Theorem | $9ms$ |
| Prop 3.2 2. | $\forall s,t.(\mathbf{T} \in s \vee \mathbf{T} \in t) \Leftrightarrow \mathbf{T} \in s \sqcup_t t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{B} \in s \vee \mathbf{B} \in t) \Leftrightarrow \mathbf{B} \in s \sqcup_t t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{F} \in s \wedge \mathbf{F} \in t) \Leftrightarrow \mathbf{F} \in s \sqcup_t t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{N} \in s \wedge \mathbf{N} \in t) \Leftrightarrow \mathbf{N} \in s \sqcup_t t$ | Theorem | $9ms$ |
| Prop 3.2 3. | $\forall s,t.(\mathbf{T} \in s \wedge \mathbf{T} \in t) \Leftrightarrow \mathbf{T} \in s \sqcup_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{N} \in s \wedge \mathbf{N} \in t) \Leftrightarrow \mathbf{N} \in s \sqcup_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{F} \in s \vee \mathbf{F} \in t) \Leftrightarrow \mathbf{F} \in s \sqcup_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{B} \in s \vee \mathbf{B} \in t) \Leftrightarrow \mathbf{B} \in s \sqcup_f t$ | Theorem | $8ms$ |
| Prop 3.2 4. | $\forall s,t.(\mathbf{T} \in s \vee \mathbf{T} \in t) \Leftrightarrow \mathbf{T} \in s \sqcap_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{N} \in s \vee \mathbf{N} \in t) \Leftrightarrow \mathbf{N} \in s \sqcap_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{F} \in s \wedge \mathbf{F} \in t) \Leftrightarrow \mathbf{F} \in s \sqcap_f t$ | Theorem | $8ms$ |
| | $\forall s,t.(\mathbf{B} \in s \wedge \mathbf{B} \in t) \Leftrightarrow \mathbf{B} \in s \sqcap_f t$ | Theorem | $8ms$ |
| Def 3.6 1. | $\forall a,b.a \leq_t b \Rightarrow -_t b \leq_t -_t a$ | Theorem | $421ms$ |
| | $\forall a,b.a \leq_f b \Rightarrow -_t a \leq_f -_t b$ | Theorem | $422ms$ |
| | $\forall a,b.a \leq_i b \Rightarrow -_t a \leq_i -_t b$ | Theorem | $8ms$ |
| | $\forall a. -_t -_t a = a$ | Theorem | $15ms$ |
| Def 3.6 2. | $\forall a,b.a \leq_t b \Rightarrow -_f a \leq_t -_f b$ | Theorem | $419ms$ |
| | $\forall a,b.a \leq_f b \Rightarrow -_f b \leq_f -_f a$ | Theorem | $423ms$ |
| | $\forall a,b.a \leq_i b \Rightarrow -_f a \leq_i -_f b$ | Theorem | $9ms$ |
| | $\forall a. -_f -_f a = a$ | Theorem | $17ms$ |

Table 3: Automated verification results of soundness relevant properties from Shramko and Wansing's monograph [33]. The time results refer to the measurements with Satallax 2.7.

# 6 Conclusion

Various techniques to automate reasoning in many-valued logics have been presented in the literature [26, 3]. The approach presented here, which employs a semantic embedding in classical higher-order logic, provides a theoretically and pragmatically appealing alternative. In particular, it is readily applicable (with off the shelf higher-order reasoners), it enables object-level and meta-level reasoning and it supports further logic extensions and combinations.

Various extensions of many-valued logics have been studied in the literature. Examples include many-valued modal logics or many-valued predicate logics.

Respective extensions of our embedding of *SIXTEEN* in HOL are analogously feasible. In particular, it should be possible to adapt the embedding of quantified modal logics [10] and combine it with the work presented here. Shramko and Wansing [33, pp.216], for example, present an idea to develop first-order trilattice logics from modal trilattice logics. In this context, a Kripke-style semantics for quantification is provided in the following form:

$$M, \alpha \models \forall x A \text{ iff for every state } \beta: \text{ if } \alpha R_x \beta, \text{ then } M, \beta \models A$$

In previous work [10] we have illustrated that similar clauses (e.g. the modal box operator) can easily be encoded. Here, the accessibility relation $R_x$ depends on the individual $x$, but such a dependency can easily be captured.

Further future work includes the application of the presented automation technique to more practically motivated examples. We are confident that this approach can indeed be used to deal with meaningful reasoning tasks where e.g. linguistic vagueness or uncertainty is involved. Also the meta-level reasoning capabilities of our approach leave room for much further work. In fact, we are positive that many meta-level statements and theorems in textbooks and publications can at least partially be verified (or falsified) with it.

Moreover, it should be possible to provide human-intuitive proof tactics in proof assistants to support interactive proof development. It was shown in previous work that similar tactics for modal logic could successfully be employed in such proof assistants [16]. In combination with proof automation, this should lead to fruitful employment for computer-aided argumentation and reasoning within theoretical philosophy.

## Acknowledgment

We thank the anonymous reviewers for their very helpful comments. Moreover, we are grateful for the comments we have received by Yaroslav Shramko and Heinrich Wansing. This work has been supported by the German National Research Foundation (DFG) under grants BE 2501/11-1 (Leo-III) and BE2501/9-2 (Ontoleo).

## References

[1] P. B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37(2):395–397, 1972.

[2] P. B. Andrews. General models, descriptions, and choice in type theory. *Journal of Symbolic Logic*, 37(2):385–394, 1972.

[3] M. Baaz, C. Fermller, and G. Salzer. *Handbook of Automated Reasoning*, chapter Automated Deduction in Many-Valued Logics. Elsevier, 2001.

[4] H. P. Barendregt, W. Dekkers, and R. Statman. *Lambda Calculus with Types*. Perspectives in logic. Cambridge University Press, 2013.

[5] N. D. Belnap. A useful four-valued logic. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, volume 2 of *Episteme*, pages 5–37. Springer NL, 1977.

[6] N. D. Belnap. How a computer should think. In A. Anderson, N. D. Belnap, and J. M. Dunn, editors, *Entailment: The Logic of Relevance and Necessity*, volume 2. Princeton University Press, 1992.

[7] C. Benzmüller. Invited Talk: On a (Quite) Universal Theorem Proving Approach and Its Application in Metaphysics. In Hans De Nivelle, editor, *TABLEAUX 2015*, volume 9323 of *LNAI*, pages 209–216, Wroclaw, Poland, 2015. Springer.

[8] C. Benzmüller, D. Gabbay, V. Genovese, and D. Rispoli. Embedding and automating conditional logics in classical higher-order logic. *Annals of Mathematics and Artificial Intelligence*, 66(1-4):257–271, 2012.

[9] C. Benzmüller and D. Miller. Automation of Higher-Order Logic. In Dov M. Gabbay, Jörg H. Siekmann, and John Woods, editors, *Handbook of the History of Logic, Volume 9 — Computational Logic*, pages 215–254. North Holland, Elsevier, 2014.

[10] C. Benzmüller and L. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue on Multimodal Logics)*, 7(1):7–20, 2013.

[11] C. Benzmüller, L. C. Paulson, N. Sultana, and F. Theiß. The higher-order prover LEO-II. *Journal of Automated Reasoning*, 55(4):389–404, 2015.

[12] C. Benzmüller and D. Scott. Automating free logic in Isabelle/HOL. In G.-M. Greuel, T. Koch, P. Paule, and A. Sommese, editors, *Mathematical Software – ICMS 2016, 5th International Congress, Proceedings*, volume 9725 of *LNCS*, pages 43–50, Berlin, Germany, 2016. Springer.

[13] C. Benzmüller, L. Weber, and B. Woltzenlogel Paleo. Computer-Assisted Analysis of the Anderson-Hájek Ontological Controversy. In Ricardo Souza Silvestre and Jean-Yves Béziau, editors, *Handbook of the 1st World Congress on Logic and Religion, Joao Pessoa, Brasil*, pages 53–54, 2015.

[14] C. Benzmüller and B. Woltzenlogel Paleo. Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 93 – 98. IOS Press, 2014.

[15] C. Benzmüller and B. Woltzenlogel Paleo. Higher-order modal logics: Automation and applications. In Adrian Paschke and Wolfgang Faber, editors, *Reasoning Web 2015*, number 9203 in LNCS, pages 32–74, Berlin, Germany, 2015. Springer.

[16] C. Benzmüller and B. Woltzenlogel Paleo. Interacting with modal logics in the coq proof assistant. In Lev D. Beklemishev and Daniil V. Musatov, editors, *Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*, volume 9139 of *LNCS*, pages 398–411. Springer, 2015.

[17] J. C. Blanchette and T. Nipkow. *Nitpick: A Counterexample Generator for Higher-Order Logic Based on a Relational Model Finder*, pages 131–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[18] C. E. Brown. Satallax: An Automatic Higher-Order Prover. In *Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, pages 111–117, 2012.

[19] A. Church. A Set of Postulates for the Foundation of Logic. *Annals of Mathematics*, 33(2):346–366, 1932.

[20] A. Church. A Set of Postulates for the Foundation of Logic, Second Paper. *The Annals of Mathematics*, 34(4):839–864, October 1933.

[21] A. Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5(2), 1940.

[22] Michael Dummett. A propositional calculus with denumerable matrix. *The Journal of Symbolic Logic*, 24:97–106, 6 1959.

[23] Melvin Fitting. Kleene's logic, generalized. *Journal of Logic and Computation*, 1(6):797–810, 1991.

[24] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

[25] K. Gödel. Zum intuitionistischen aussagenkalkül. *Ergebnisse eines mathematischen Kolloquiums*, 4:34–38, 1933.

[26] R. Hähnle. *Automated deduction in multiple-valued logics.* Clarendon Press, Oxford, 1993.

[27] P. Hajek. *Metamathematics of fuzzy logic.* Dordrecht: Kluwer, 1998.

[28] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 06 1950.

[29] J. Łukasiewicz. O logice trojwartosciowej. *Ruch Filozoficzny*, 5:170–171, 1920.

[30] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[31] S. P. Odintsov. On Axiomatizing Shramko-Wansings Logic. *Studia Logica*, 91(3):407–428, 2009.

[32] Y. Shramko and H. Wansing. Some Useful 16-Valued Logics: How a Computer Network Should Think. *Journal of Philosophical Logic*, 34(2):pp. 121–153, 2005.

[33] Y. Shramko and H. Wansing. *Truth and Falsehood: An Inquiry into Generalized Logical Values*. Trends in Logic. Springer Netherlands, 2011.

[34] Yaroslav Shramko, J. Michael Dunn, and Tatsutoshi Takenaka. The trilattice of constructive truth values. *Journal of Logic and Computation*, 11(6):761–788, 2001.

[35] A. Steen and M. Wisniewski. Embedding of First-Order Nominal Logic into HOL. In J.-Y. Beziau, S. Ural, A. Buchsbaum, I. Tasdelen, and V. Kamer, editors, *5th World Congress and School on Universal Logic (UNILOG'15)*, pages 337–339, Istanbul, Turkey, 2015.

[36] A. Steen, M. Wisniewski, and C. Benzmüller. Agent-based HOL reasoning. In G.-M. Greuel, T. Koch, P. Paule, and A. Sommese, editors, *Mathematical Software – ICMS 2016, 5th International Congress, Proceedings*, volume 9725 of *LNCS*, Berlin, Germany, 2016. Springer. To appear.

[37] G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.

[38] G. Sutcliffe and C. Benzmüller. Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010.

[39] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Alexander Steen
Freie Universität Berlin
Institute of Computer Science
a.steen@fu-berlin.de

Christoph Benzmüller
Stanford University
CSLI/Cordula Hall
c.benzmueller@gmail.com