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

Alexander Steen and Christoph Benzmüller

\{a.steen,c.benzmueller\}@fu-berlin.de

Freie Universität Berlin, Germany

Introduction. Classical logics are based on the bivalence principle, that is, the set of truth-values \( V \) has cardinality \(|V| = 2\), usually with \( V = \{T, F\} \) where \( T \) and \( F \) stand for truthhood and falsity, respectively. Many-valued logics generalize this requirement to more or less arbitrary sets of truth-values, rather referred to as *truth-degrees* in that context. Popular examples of many-valued logics are Gödel logics, Łukasiewicz logics or fuzzy logics with denumerable (or even larger in the case of fuzzy logic) sets of truth-degrees, and, from the class of finitely-many-valued logics, Dunn/Belnap’s four-valued logic [12].

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

This work presents an approach for automating a sixteen-valued logic denoted *SIXTEEN*. This logic has been developed by Shramko and Wansing as a generalization of the mentioned four-valued system to knowledge bases in computer networks [9] and was subsequently further investigated in various contexts (e.g. [8,10]). In *SIXTEEN*, the truth-degrees are given by the power set of Belnap’s truth values, i.e.

\[
V = 2^{\{N,T,F,B\}} = \{N, N, T, F, B, \ldots, \{N, T, F, B\}\}
\]

where \( N, T, F \) and \( B \) are the respective singleton sets containing \( N, T, F \) and \( B \). The remaining truth-degrees are named using a combination of the letters \( N, T, F \) and \( B \), representing the truth-degree that contains the respective elements when regarded as a set (e.g. \( N T \) for the set \( \{N, T\} \)). 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 [10], to which we refer to for a thorough investigation of *SIXTEEN*.

* This work has been supported by the German National Research Foundation (DFG) under grant BE 2501/11-1 (Leo-III).
Our Approach. In order to offer computer assisted reasoning in SIXTEEN, we employ a semantic embedding into classical higher-order logic (here used synonymous to Church’s Simple Theory of Types), denoted HOL [3]. HOL is an expressive formal system based on a typed $\lambda$-calculus that allows quantification over arbitrary sets and functions. Its semantics is meanwhile well-understood and several sophisticated automated theorem provers for HOL – with respect to Henkin semantics [6] – exist. Its syntax is given by

$$s, t ::= X_\tau \mid c_\tau \mid (\lambda X_\tau. \ s_\tau)_\nu \mid (s_\nu \ t_\tau)_\nu$$

where $\tau, \nu$ are types, $\nu \tau$ denotes the type of functions from arguments of type $\tau$ to values of type $\nu$, and $\circ$ denotes the type of (classical) truth-values. The constants $c_\tau$ come from a signature $\Sigma$ which we choose to consist at least of the logical operators for negation, disjunction, and universal quantification for each type $\tau$, i.e. $\neg$, $\lor$, and $\forall^\tau \sigma$ for each type $\tau$, respectively. The remaining connectives can be defined in the usual way. For convenience we allow infix notation for the common logical connective, i.e. $s \circ t$ instead of $((s \circ t) t)$.

In HOL, a set $M$ can be modeled by its characteristic function $\chi_M$ that is true for any argument $m \in M$ and false otherwise. Using this encoding, we model the sixteen truth-values of SIXTEEN by (we omit types if possible)

$$\begin{array}{lcl}
N & = & \lambda n_{oo}. F \\
N & = & \lambda n_{oo}. \neg n F \land \neg n T \\
T & = & \lambda n_{oo}. \neg n F \land n T \\
T & = & \lambda n_{oo}. n F \land n T \\
F & = & \lambda n_{oo}. n F \land \neg n T \\
F & = & \lambda n_{oo}. \neg n F \land n T \\
B & = & \lambda n_{oo}. n F \land n T \\
B & = & \lambda n_{oo}. n F \land n T \\
N \circ F & = & \lambda n_{oo}. \neg n F \land \neg n T \\
N \circ F & = & \lambda n_{oo}. n F \land \neg n T \\
N B & = & \lambda n_{oo}. n F \land \neg n T \\
N B & = & \lambda n_{oo}. n F \land \neg n T \\
N B & = & \lambda n_{oo}. n F \land \neg n T \\
N B & = & \lambda n_{oo}. n F \land \neg n T \\
\end{array}$$

The three distinct ordering relations on truth-degrees, denoted $\leq_t$, $\leq_f$, $\leq_i$, order by truthhood, falsehood and entropy. We now present our encoding of the logical operations corresponding to truthhood reasoning, i.e. based on $\leq_t$ (cf. [10] for details). The remaining operations have also been encoded, but are omitted here.

The definition of truthful subsets $(\cdot)_t^t$ and truthless subsets $(\cdot)_t^{-t}$ of truth-degrees can be defined in a straight-forward way:

$$(v)_t^t := \lambda n_{oo}. (v n) \land (n T) \quad (v)_t^{-t} := \lambda n_{oo}. (v n) \land \neg (n T)$$

Further relevant embedded definitions include

$$\begin{array}{l}
\leq_t := \lambda v_{oo}(o). \lambda w_{oo}(o). \forall n_{oo}. ((v^t n) \supset (w^t n)) \land ((w^{-t} n) \supset (v^{-t} n)) \\
\sqcup_t := \lambda v_{oo}(o). \lambda w_{oo}(o). v^t \cup w^t \cup (w^{-t} \land v^{-t}) \\
\sqcap_t := \lambda v_{oo}(o). \lambda w_{oo}(o). v^t \cup w^t \cup (w^{-t} \land v^t) \\
\sim_t := \lambda v_{oo}(o). \lambda n_{oo}. (v (\lambda b_\circ (-b \supset n F) \land (b \supset \neg (n T))))
\end{array}$$

First experiments. The semantic embedding has successfully been employed in two different versions\(^1\) that is, as theory for the interactive theorem prover

\(^1\) The embedding and experiment files can be found at [http://inf.fu-berlin.de/~lex/sixteen.tar](http://inf.fu-berlin.de/~lex/sixteen.tar)
Isabelle/HOL \cite{Nipkow2002} and as THF axiomatization \cite{SutcliffeBenzmueller2010} ready to use with any TPTP syntax \cite{Sutcliffe2009} compliant automated theorem prover for HOL. In the latter case, the experiments have been conducted using the two provers LEO-II \cite{BenzmuellerTheissPaulsonFietzke2008} and Satallax \cite{Brown2012}. In these experiments, we were able to automatically verify several meta-logical properties about SIXTEEN (cf. Prop. 3.2, Prop. 3.4 and Def. 3.6 from \cite{ShramkoWansing2011}). These properties specify the behaviour of the logical connectives of SIXTEEN, where most of them could automatically be verified in under 10ms.

One example of such a proposition is (taken from Prop. 3.2):

\[
\forall v_0, w_0. (\forall (N \in v) \land (N \in w)) \iff N \in (v \sqcup w)
\]

Using the embedding technique, common higher-order provers can be utilized for reasoning in SIXTEEN (and potentially many further logics), where otherwise a special-purpose reasoner would need to be developed (for each individual logic). Additionally, meta-logical reasoning is freely available using the HOL meta-logic. Further work includes the application of the presented automation technique to more practically motivated examples. We are positive that this approach can indeed be used to deal with meaningful reasoning tasks where e.g. linguistic vagueness or uncertainty is involved.

References