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 $V = \{\mathrm{T}, \mathrm{F}\}$ 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 [1,2].

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 $\{\mathrm{T}, \mathrm{F}\}$, i.e. $V = \{\mathbf{N}, \mathbf{T}, \mathbf{F}, \mathbf{B}\}$, where $\mathbf{N}$ denotes the empty set (mnemonic for *None*), $\mathbf{T}$ and $\mathbf{F}$ the singleton sets of the respective classical truth-value, and $\mathbf{B}$ the set $\{\mathrm{T}, \mathrm{F}\}$ (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.

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

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 $\{\mathbf{N}, \mathbf{T}\}$). 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 [5]. HOL is an expressive formal system based on a typed λ-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

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

where τ, ν are types, $\nu\tau$ denotes the type of functions from arguments of type τ to values of type ν, and o denotes the type of (classical) truth-values. The constants c_τ come from a signature Σ which we choose to consist at least of the logical operators for negation, disjunction, and universal quantification for each type τ, i.e. $\neg_{oo}$,$\vee_{ooo}$ and $\Pi^\tau_{o(o\tau)}$ for each type τ, respectively. The remaining connectives can be defined in the usual way. For convenience we allow infix notation for the common logical connective, i.e. $s \vee t$ instead of $((\vee\ s)\ t)$.

In HOL, a set M can be modeled by its characteristic function χ_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)

$\mathbf{N} = \lambda n_{oo}.\ F$	$\mathbf{FT} = \lambda n_{oo}.\ (n\ F \wedge \neg n\ T) \vee (\neg n\ F \wedge n\ T)$
$\boldsymbol{N} = \lambda n_{oo}.\ \neg n\ F \wedge \neg n\ T$	$\mathbf{FB} = \lambda n_{oo}.\ n\ F$
$\boldsymbol{T} = \lambda n_{oo}.\ \neg n\ F \wedge n\ T$	$\mathbf{TB} = \lambda n_{oo}.\ n\ T$
$\boldsymbol{F} = \lambda n_{oo}.\ n\ F \wedge \neg n\ T$	$\mathbf{NFT} = \lambda n_{oo}.\ \neg n\ F \vee \neg n\ T$
$\boldsymbol{B} = \lambda n_{oo}.\ n\ F \wedge n\ T$	$\mathbf{NFB} = \lambda n_{oo}.\ n\ F \vee \neg n\ T$
$\mathbf{NF} = \lambda n_{oo}.\ \neg n\ T$	$\mathbf{NTB} = \lambda n_{oo}.\ \neg n\ F \vee n\ T$
$\mathbf{NT} = \lambda n_{oo}.\ \neg n\ F$	$\mathbf{FTB} = \lambda n_{oo}.\ n\ F \vee n\ T$
$\mathbf{NB} = \lambda n_{oo}.\ (\neg n\ F \wedge \neg n\ T) \vee (n\ F \wedge n\ T)$	$\mathbf{A} := \mathbf{NFTB} = \lambda n_{oo}.\ T$

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 $(.)^t$ and *truthless* subsets $(.)^{-t}$ of truth-degrees can be defined in a straight-forward way:

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

Further relevant embedded definitions include

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ ((v^t\ n) \supset (w^t\ n)) \wedge ((w^{-t}\ n) \supset (v^{-t}\ n))$$
$$\sqcup_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^t \cup w^t \cup \left(w^{-t} \cap v^{-t}\right)$$
$$\sqcap_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^{-t} \cup w^{-t} \cup \left(w^t \cap v^t\right)$$
$$\sim_t := \lambda v_{o(oo)}.\lambda n_{oo}.\ (v\ (\lambda b_o.(\neg b \supset n\ \mathrm{F}) \wedge (b \supset \neg(n\ \mathrm{T}))))$$

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`

Isabelle/HOL [7] and as THF axiomatization [12] ready to use with any TPTP syntax [11] compliant automated theorem prover for HOL. In the latter case, the experiments have been conducted using the two provers LEO-II [3] and Satallax [4]. 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 [10]). 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):

$$\forall v_{o(oo)}.\forall w_{o(oo)}.\ ((\mathbf{N} \in v) \wedge (\mathbf{N} \in w)) \longleftrightarrow \mathbf{N} \in (v \sqcup_t w)$$

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

1. 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.
2. 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.
3. C. Benzmüller, F. Theiss, L. Paulson, and A. Fietzke. LEO-II – A Cooperative Automatic Theorem Prover for Higher-Order Logic (System Description). In *Automated Reasoning, IJCAR 2008, Sydney, Australia, Proceedings*, volume 5195 of *LNCS*, pages 162–170. Springer, 2008.
4. Chad 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.
5. A. Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5(2), 1940.
6. L. Henkin. Completeness in the theory of types. *J. Symb. Log.*, 15(2):81–91, 1950.
7. T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.
8. S. P. Odintsov. On Axiomatizing Shramko-Wansings Logic. *Studia Logica*, 91(3):407–428, 2009.
9. 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.
10. Y. Shramko and H. Wansing. *Truth and Falsehood: An Inquiry into Generalized Logical Values*. Trends in Logic. Springer Netherlands, 2011.
11. 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.
12. 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.