Meta-Logical Reasoning in Higher-Order Logic

Villadsen, Jørgen; Schlichtkrull, Anders; Hess, Andreas Viktor

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Villadsen, J., Schlichtkrull, A., & Hess, A. V. (2015). *Meta-Logical Reasoning in Higher-Order Logic*. Poster session presented at LOGICA 2015 - 29th Annual International Symposia Devoted to Logic, Hejnice, Czech Republic.

Meta-Logical Reasoning in Higher-Order Logic

Jørgen Villadsen, Anders Schlichtkrull & Andreas Viktor Hess
Technical University of Denmark - DTU Compute

Abstract

The semantics of first-order logic (FOL) can be described in the meta-language of higher-order logic (HOL). Using HOL one can prove key properties of FOL such as soundness and completeness. Furthermore, one can prove sentences in FOL valid using the formalized FOL semantics. To aid in the construction of the proof an interactive proof assistant like Isabelle can be used. The proof assistant can even automate simple proofs using the formalized FOL semantics.

Introduction

In textbooks the language of first-order logic (FOL) is usually presented in English. FOL is the object language, since it is the logic that is described, and English is the meta-language, since it is the language that describes the object language. However, instead of using English as meta-language we can also use a meta-logic. We will show how higher-order logic (HOL) can be used as a meta-logic to describe and reason about FOL, which is possible as HOL is much more expressive than FOL, cf. Farmer (2008).

Logic itself is about formalizing which arguments are valid. Thus in FOL we have a clear definition of which theorems are valid and which are not. However, it is also interesting to prove theorems *about* FOL, for instance the soundness and completeness of a proof system for FOL. By using HOL as meta-language we can ensure that there is also a clear definition of which theorems about FOL are valid, cf. Harrison (1998). Furthermore, we can show the theorems about FOL to be valid by proving them in a sound proof system for HOL.

Proof systems for HOL have been implemented in interactive proof assistants which are computer programs that can help their users in proving theorems. In the following we will use the Isabelle proof assistant, cf. Nipkow (2002). In addition to helping the users to construct correct proofs, the proof assistants can in some cases even do the proofs automatically.

Formalization in Isabelle

We consider a formalization in Isabelle of FOL with only binary predicates:

```
theory Semantics imports Main begin

type_synonym id = string

datatype tm = Var nat | Con id
datatype fm = Falsity | Pre id tm tm | Imp fm fm | Uni fm

primrec val :: "(nat => 'u) => (id => 'u) => tm => 'u" where
"val e f (Var v) = e v" |
"val e f (Con c) = f c"

primrec sem :: "(nat => 'u) => (id => 'u) => (id => 'u * 'u => bool) => fm => bool" where
"sem e f g Falsity = False" |
"sem e f g (Pre s a b) = g s (val e f a, val e f b)" |
"sem e f g (Imp p q) = (if sem e f g p then sem e f g q else True)" |
"sem e f g (Uni p) = (!x. sem (%n. if n=0 then x else e (n - 1)) f g p)"

end
```

The terms and formulas of the FOL language are defined as the datatypes `tm` and `fm`, respectively. Variables are indexed using de Bruijn indices. The semantics of the language is defined using the function `sem` where `e` is the environment, i.e. a mapping of variables to elements of the universe `'u`, `f` maps constants to elements of `'u`, and `g` gives the semantics of the predicates. Most of the cases of `sem` should be self-explanatory, but the `Uni` case is complicated. The details are not important here, but it uses the universal quantifier (`!`) to consider all values of the universe `'u`. It also uses the lambda operator (`%`) to keep track of the indices of the variables.

Examples of Meta-Logical Reasoning

By using HOL as the meta-language for FOL it is possible to make use of proof assistants such as Isabelle to reason about certain properties of FOL. For example, let `syn` be a proof system for FOL implemented as a predicate in Isabelle of type `fm => bool` (an inductive definition). If it is sound then we can prove this in Isabelle as the theorem `syn fml ==> !e f g. sem e f g fml`. Likewise completeness `(!e f g. sem e f g fml) ==> syn fml` can be proved if the proof system is indeed complete. For example, in Berghofer (2007) a natural deduction proof system for FOL is proven sound and complete. A tool for teaching logic based on natural deduction has recently been developed and proved sound by Villadsen (2015).

Meta-logical reasoning using the formalized semantics also enables formal proofs of sentences using only the semantics of the object language. For example, we wish to show that the sentence $\forall x.\forall y.\, P(y,x) \rightarrow P(y,x)$ is valid. We can prove this by first fixing two arbitrary elements u, w of the universe and show $P(y,x) \rightarrow P(y,x)$ for an arbitrary environment updated to map variable x to u and y to w, and then use the fact that any denotation g of predicates maps (P, u, w) to either true or false. In both cases $P(y,x) \rightarrow P(y,x)$ holds. This proof sketch can then be extended to a readable proof in Isabelle. The sentence can even be proved automatically in Isabelle using the formalized semantics:

```
theorem "!e f g. sem e f g (Uni (Uni (Imp (Pre ''P'' (Var 0) (Var 1))
                                          (Pre ''P'' (Var 0) (Var 1)))))"
by auto
```

On the other hand the sentence $\forall x.\forall y.\, P(x,y) \rightarrow P(y,x)$ is not valid. Therefore we can ask Isabelle to search for a counterexample, using the `nitpick` command, and it is able to find one.

Finally, the approach we have presented can even be used with HOL as the object language, that is, the semantics of HOL and a proof system for HOL can also be formalized in HOL, most extensively by Kumar (2014), although of course self-verification is not possible due to the second incompleteness theorem. Paulson (2014) has recently formalized the incompleteness theorems in Isabelle.

References

Berghofer, S. (2007). First-Order Logic According to Fitting. Archive of Formal Proofs.

Harrison, J. (1998). Formalizing Basic First Order Model Theory. Lecture Notes in Computer Science 1497, 153-170, Springer.

Farmer, W. M. (2008). The Seven Virtues of Simple Type Theory. Journal of Applied Logic, 6(3), 267-286.

Kumar, R., R. Arthan, M. O. Myreen & S. Owens (2014). HOL with Definitions: Semantics, Soundness, and a Verified Implementation. Lecture Notes in Computer Science 8558, 308-324, Springer.

Nipkow, T., L. C. Paulson & M. Wenzel (2002). Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Lecture Notes in Computer Science 2283 Springer.

Paulson, L. C. (2014). A Machine-Assisted Proof of Gödel's Incompleteness Theorems for the Theory of Hereditarily Finite Sets. Review of Symbolic Logic 7(3), 484-498.

Villadsen, J., A. B. Jensen & A. Schlichtkrull (2015). NaDeA: A Natural Deduction Assistant with a Formalization in Isabelle. Accepted at 4th International Conference on Tools for Teaching Logic, Rennes, France.