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

Link back to DTU Orbit

Citation (APA):

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Meta-Logical Reasoning in Higher-Order Logic

Jørgen Villadsen, Anders Schlichtkrull & Andreas Viktor Hess

Technical University of Denmark - DTU Compute

Abstract

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 fm ⇒ !e f g. sem e f g fm. Likewise completeness (!e f g. sem e f g fm) ⇒ syn fm 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 ∀x.∀y. P(x,y) → 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) → 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 q of predicates maps (P, u, w) to either true or false. In both cases P(y,x) → 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:

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

On the other hand the sentence ∀x.∀y. P(x,y) → 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


