On the Foundation of Isabelle/HOL *

Arve Gengelbach
arve.gengelbach@it.uu.se

Dept. of Information Technology, Uppsala University, Sweden

Abstract. Interactive theorem provers are used to allow formalisation of mathematical proofs into *theories*, written in a precise language of definitions and proofs. Such a theory is step-wise extended by constant instance definitions and type definitions and contains proofs of theorems that use these definitions. If at any point a theory is inconsistent, i.e. every formula is deducible, the formalisation effort was futile. We set out to extend the existing foundational work on the model-theoretic and proof-theoretic foundation of such definitional theories, as used in the theorem prover Isabelle/HOL. For this logical system we study semantics that entail completeness and soundness, which are the properties that link the model-theoretic and the proof-theoretic perspectives. We formalize the obtained results in the theorem prover Isabelle/HOL to strengthen the confidence in correctness of our work.

1 Problem

Interactive theorem provers support a user proving theorems in different logics. Well-known provers are based on higher-order logic or on constructive type theory. Especially for technical proofs, in addition to pen-and-paper proofs, theorem provers provide an appealing framework for formalisation. Not only is a formalisation interesting from a theoretic point of view but it also is practically applicable: For example Isabelle allows to export code (via so called code generators [4]) whose behaviour is proven to match the abstract-level specification by abstract-level theorems. It has been applied for different verification efforts, exemplarily the verification of the single processor behaviour of the seL4 microkernel [5].

All verification efforts are of little use if the theoretical foundation of the framework is poorly understood. At its core Isabelle/HOL consists of a minimal theory with a mechanism to extend a theory by definitions of constants and types. These theories are called *definitional theories* [10]. If both the smallest theory – called *minimal theory* – and the extension mechanism are properly designed the system will disallow that any statement is derivable. For the minimal theory we require its *consistency* and the consistency of any theory extension, i.e. from a theory a proof of False is not possible or equivalently some statements are not derivable. Furthermore we require theory extension to be *proof-theoretic*

* The author is expected to graduate from PhD studies in 2020.

conservative (also *syntactically conservative*) [12, p. 41], viz. for a theory T and an extension T', any formula in the language of T which is provable from the extension T' is also provable from the smaller theory T. This syntactic conservativity implies consistency of all theory extensions. An extension by definitions shall be syntactic, that is defined terms in a formula of an extended theory can be replaced by their defining terms and regarded as formulas of smaller theories. This so-called *realizability* together with syntactic conservativity are summarized as *meta-safe* extensions [13].

In meta-safe theory extensions, definitions can be unfolded and understood as syntactic abbreviations. In the context of theorem provers this is interesting as theorems of meta-safe theory extensions can be expressed and proven equivalently in a smaller theory for a possible benefit of reduction of complexity.

Despite regarding a theory and syntactic deducibility of formulae according to deduction rules, its models and evaluation of formulae in a model offer another method of study. A model defines what is valid, i.e. what evaluates to True, especially a model of a theory is a model that at least renders all axioms of the theory valid. A model of a smaller theory T in a signature Σ can be extended to a model of a larger theory, with a larger signature such that all valid formulae of the smaller model are valid in the larger model. This allows us to regard a dual notion of proof-theoretic conservativity, called *model-theoretic* conservativity: A theory extension $T \subseteq T'$ is model-theoretic conservative if each model $\mathcal{M}$ for T can be extended to a model $\mathcal{M}$ for T' (i.e. $\mathcal{M}'$ models T' and $\mathcal{M}$ and $\mathcal{M}'$ agree on the interpretations of terms over the signature Σ).

Both the semantic and the syntactic perspective of a deduction system can be combined: If for a theory any deducible formula implies its validity in any of the theory's models, the system is called *sound*; the converse is called *complete*. In a sound and complete system model-theoretic conservativity implies proof-theoretic conservativity.

Isabelle/HOL implements higher-order logic with *rank-1-polymorphism* and *ad hoc* overloading. That is, for a constant several definitions for different non-overlapping polymorphic types (i.e. non-unifying types) can be provided. An example illustrates these features. We define a type $\alpha\ \mathsf{lists}$ and constant instances map, that applies a given function to every item in a given structure, for lists and for sets. We assume that sets have been defined. Lists shall be defined inductively as either the empty list $[\,]$ or as $x : xs$ with a head x of type α and a tail xs of type $\alpha\ \mathsf{list}$. We extend this theory by an operation on lists: $\mathsf{map}_{(\alpha\to\beta)\to\alpha\ \mathsf{list}\to\beta\ \mathsf{list}}$. On empty lists this is defined as $\mathsf{map}(f, [\,]) \equiv [\,]$ and $\mathsf{map}(f, x : xs) \equiv f(x) : \mathsf{map}(f, xs)$ for non-empty lists. We introduce a constant instance for sets $\mathsf{map}_{(\alpha\to\beta)\to\alpha\ \mathsf{set}\to\beta\ \mathsf{set}}$ by defining $\mathsf{map}(f, A) \equiv f(A)$. By these definitions lists and sets are not instances of one another and the definitions are not circular.

A mechanism of theory extension has to prevent overlapping definitions for the same constant and also prevent the introduction of circular dependencies. An example [6] illustrates the danger. Let c_α be a declared polymorphic constant. Let $\tau \equiv \{\mathsf{True}, c_{\mathsf{bool}}\}$ define a type and then $c_{\mathsf{bool}} \equiv \neg(\forall x_\tau, y_\tau : x_\tau = y_\tau)$ define

the constant instance. Assume that $c_{\mathsf{bool}} = \mathsf{True}$, which by the definition of τ is equivalent to $\tau \equiv \{\mathsf{True}\}$. The formula $\forall x_\tau, y_\tau : x_\tau = y_\tau$ holds (i.e. is True) as it states that τ is a singleton, and thus $c_{\mathsf{bool}} = \mathsf{False}$. Summarised, $c_{\mathsf{bool}} = \mathsf{True} \Leftrightarrow c_{\mathsf{bool}} = \mathsf{False}$ proves this theory inconsistent. Note that a declared constant c_α can be used prior to definition of any of its instances. In [6] the authors propose a mechanism for extension of definitional theories that disallows circularities and makes the extension consistent.

To achieve a solid theoretical foundation for Isabelle/HOL **we investigate if the definitional mechanism introduced in [6] makes theory extension proof-theoretic and model-theoretic conservative**. The HOL system is not complete with respect to standard semantics by an argument that bases on Gödels incompleteness theorem [10, Section 2.4.5], which motivates the study of different semantics that render the system sound and complete.

2 Related work

The documentation of the HOL system logic [10] defines and discusses the foundation of the HOL deduction system. Exemplary is the proof that the HOL system possesses a standard model for the extension mechanism: Extension by type and constant instance specification. Isabelle/HOL extends this mechanism and allows several definitions of instances for a constant and constants that need not be defined for all types.

Wenzel [13] defines safe extension of theories by constant instances, where constant instances are defined at once and extensions by type definitions and constant instance definitions can not be mixed.

Obua [11] discovers that checking conservative overloading in a logic *HOLCO* is not semi-decidable and furthermore discusses that the term rewriting system for definitions needs to be terminating for a theory to be consistent. Our initially given example shows the weak point: Inconsistency can be introduced by the interplay of type definitions and constant instance definitions, which had not been considered.

Kunčar and Popescu in [6] introduce a decidable dependency relation for definitional theories of HOL and thereby define *well-formed* definitional theories. The type substitutive transitive closure of the dependency relation of these theories is terminating, i.e. does not contain cycles. To ease the understanding we give a simplified definition: A term and either a constant instance or a type are in the type substitutive closure of the dependency relation $\rightsquigarrow^{\downarrow+}$, if the definition of the constant instance or type is necessary to evaluate the term. In addition they prove consistency of these theories by new semantics.

In [7] each definitional HOL theory is proven syntactically consistent. In a purely syntactic manner definitions can be understood as abbreviations and thus unfolded in the introduced deduction system *HOLC*. The system adds type comprehensions to translate type definitions from Isabelle/HOL into HOLC and get consistency of Isabelle/HOL by a consistency result in HOLC.

In a draft [8] the authors prove the syntactic conservativity of definitional theories as extensions of the minimal theory `MIN`. Constant instances get translated to their respective instance of the definitional term and the types are replaced in a more complex manner. This conservativity result implies consistency of any theory, by the consistency of the minimal theory.

Andrews introduces a formulation of higher-order logic $\mathcal{Q}_0$ (also known as simple type theory) [1, Chapter 5], that introduces the logical operators as constants based on equality and has one deduction rule for substitution of equal terms. The system $\mathcal{Q}_0$ is sound and complete for a non-standard semantics, that is for types α and β the domain of interpretation of functions $\mathcal{D}_{\alpha\to\beta}$ is relaxed such that it can be a proper subset of all possible values $\mathcal{D}_\alpha \to \mathcal{D}_\beta$. In contrast to HOL, $\mathcal{Q}_0$ does not support polymorphic constant definitions.

Geuvers and Nederpelt [9] and Geuvers [3] discuss properties of Calculus of Constructions extended λC with definitions and primitive notions. This flavour of typed lambda calculus with definitions, called λD_0 allows non-polymorphic definitions that are acyclic by design. Similarly to the intend in Isabelle/HOL, definitions in λD_0 are abbreviations, i.e. syntactical and can be unfolded. An extension λD additionally introduces *primitive* definitions, i.e. axioms. The authors discuss properties relating to conservativity and realizability, e.g. [9, Lem 10.4.1],[3, Lem 3.10, Lem 3.18]. The developed framework is very general and as examples for the expressiveness different logics are encoded in λD [9] and they give further evolved examples. Feasibility and decidability play an important role as λD is based on type theory.

3 Proposed solution

We propose a generalisation of [6] that equates to a notion of model-theoretic conservativity of an arbitrary extension of well-formed definitional theories. As initially discussed, model-theoretic implies proof-theoretic conservativity if soundness and completeness holds for the deduction system. We investigate whether semantics based on the ideas of [1] are sound and complete, especially the relaxation of the requirement on domains of functions. We expect the soundness result to be straight-forward, as opposed to the completeness proof, that we intend to approach by constructing a model for a consistent set of formulae [1]. Thus, we obtain an – opposed to [8] – *relative* proof-theoretic conservativity from the model-theoretic result by completeness and soundness of the proof-system.

4 Preliminary work

We have worked on model-theoretic conservativity that is based on a non-standard model definition and we have proven the following statement.

> Let D be a well-formed definitional theory with a model $\mathcal{M}$. Let D' be a well-formed definitional theory that extends D. Then there exists a model $\mathcal{M}'$ of D', such that $\mathcal{M}$ and $\mathcal{M}'$ agree on the interpretations of all terms that do not contain any instances of the terms defined by $S(D', D)$.

The mentioned theory $S(D', D)$ is the biggest subset of D' such that each of the defining terms in that set transitively uses an instance of the definitions that were added through the extension by $D' \setminus D$. The theories $D' \setminus D$ and $S(D', D)$ are equal if the terms added to D to obtain D' do not change the interpretation of the terms defined in D.

Our result [2] extends and generalises [6], as model-theoretic conservativity implies the consistency of definitional theories. Furthermore, we are working on the completeness proof that makes the proof-system complete with respect to non-standard semantics.

5 Expected contributions

We expect to contribute to the understanding of higher-order logic as implemented in Isabelle/HOL by

- the earlier sketched result on model-theoretic conservativity,
- a soundness and a completeness result with respect to non-standard semantics together with proof-theoretic conservativity (implied by the model-theoretic conservativity), and
- formalisations in Isabelle/HOL of the two expected theoretic contributions.

6 Plan for evaluation and validation

As typical for theoretic work, our results are validated by the scientific community, e.g. by peer review. Furthermore we want to validate all obtained abstract level pen-and-paper proofs by a theorem prover. The formalisation additionally strengthens the results of a pen-and-paper proof and can reveal mistakes.

7 Current status

The work on model-theoretic conservativity is accepted for publication. Until the end of this year we intend to complete the work on the soundness and completeness proof, so that it can become part of the author's licentiate thesis. The formalisation is foreseen in 2018 for the time after the licentiate. Being at an early stage in the PhD studies with planned graduation in 2020, the plan is subject to changes.

References

1. P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth through Proof*. Number 27 in Applied logic series. Kluwer Academic Publishers, Dordrecht ; Boston, 2nd ed edition, 2002.
2. A. Gengelbach and T. Weber. Model-theoretic Conservative Extension of Definitional Theories. Draft.
3. H. Geuvers. *Properties of a Lambda Calculus with Definitions*. 2014.
4. F. Haftmann and T. Nipkow. Code Generation via Higher-Order Rewrite Systems. In *SpringerLink*, pages 103–117. Springer, Berlin, Heidelberg, Apr. 2010.
5. G. Klein, J. Andronick, K. Elphinstone, G. Heiser, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. seL4: Formal Verification of an Operating-system Kernel. *Commun. ACM*, 53(6):107–115, June 2010.
6. O. Kunčar and A. Popescu. A Consistent Foundation for Isabelle/HOL. In C. Urban and X. Zhang, editors, *Interactive Theorem Proving*, number 9236 in Lecture Notes in Computer Science, pages 234–252. Springer International Publishing, Aug. 2015.
7. O. Kunčar and A. Popescu. Comprehending Isabelle/HOL's Consistency. In H. Yang, editor, *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10201 of *Lecture Notes in Computer Science*, pages 724–749. Springer, 2017.
8. O. Kunčar and A. Popescu. Safety and Conservativity of Definitions in HOL and Isabelle/HOL. Technical report, 2017.
9. R. P. Nederpelt and H. Geuvers. *Type Theory and Formal Proof: An Introduction*. Cambridge University Press, Cambridge ; New York, 2014.
10. M. Norrish and K. Slind. The HOL System LOGIC, Nov. 2014.
11. S. Obua. Checking Conservativity of Overloaded Definitions in Higher-Order Logic. In *Term Rewriting and Applications*, pages 212–226. Springer Berlin Heidelberg, Aug. 2006.
12. J. R. Shoenfield. *Mathematical Logic*. A.K. Peters, Natick, Mass, 1967.
13. M. Wenzel. Type classes and overloading in higher-order logic. In E. L. Gunter and A. Felty, editors, *Theorem Proving in Higher Order Logics*, number 1275 in Lecture Notes in Computer Science, pages 307–322. Springer Berlin Heidelberg, Aug. 1997.