Abstract Deduction and Inferential Models for Type Theory

Paolo Gentilini[a], Maurizio Martelli[b]

[a]*ANSAS Liguria, Via Assarotti 15, Genova, Italy*
and
IMATI-CNR, Via De Marini 6, 16149 Genova, Italy

[b]*DISI, University of Genova, Via Dodecaneso 35, 16146 Genova, Italy*

Abstract
An inferential semantics for full Higher Order Logic (HOL) is proposed. The paper presents a constructive notion of model, that being able to capture relevant computational aspects is particularly suited for the applications of HOL to computer science. The inferential semantics is based on the introduction of new abstract deduction structures (ADS) that express the action of the Comprehension Axiom in a higher order logic proof. The ADSs allow to define an inferential algebra of higher order proof-trees, endowed with two binary operations, the abstraction and the contraction, each consisting of constructive reductions between proofs. Typed formulas are interpreted by trees, and the reductions between proofs correspond to the logical connectives of the interpreted formula. Higher order logic is sound and complete w.r.t. the given inferential semantics.

Keywords: Logic in Computer Science, constructive semantics for higher order logic, proof-theory

1. Introduction

In this work an inferential semantics for classical Higher Order Logic is proposed which provides a new kind of syntactic models. The formulation of Higher Order Logic considered in this paper is the sequent version of Church's theory of types defined by Miller, Scedrov, Nadathur and Pfenning in the early 1990's (see, e.g., [23]), so that also typed λ-calculus is included in the system.

The main goal of the paper is to introduce a new notion of interpretation that could more easily convey a semantic characterization related to the notion of proof: *inferential semantics* introduces a class of models of Higher Order Logic with such a constructive nature, which is strong enough to allow a completeness result.

The well known papers of Henkin [17] and Andrews [1], [2] provided a semantics for Church's classical type theory by means of *general models*, where the information about the semantical features of the interpreted formula remains mostly implicit. Conversely, *inferential semantics* aims at interpreting a typed formula B_α through an effectively constructed object, including explicit information, that translates the syntax of B_α, both the logical part and the typed part, into some precise inferential properties of a sequent tree.

To this aim, a new proof-theoretic analysis of full Higher Order Logic is proposed, based on a formal characterization of the role of the Comprehension-rules ($\exists$-R, $\forall$-L, i.e. the two rules expressing the Comprehension Axiom) in a sequent tree. Such analysis allows to define an abstraction of the proof-trees resulting in sequences of formulas (called abstract deduction structures) which are linked to specific occurrences of

Comprehension-rules in the tree. Thus, essentially, inferential semantics associates to a formula B_α a set of abstract deduction structures occurring in a sequent tree.
As a consequence of the effective character of the inferential interpretation, a new notion of *meaning* for typed formulas of arbitrary type can be introduced. It is independent of the truth denotation: thus, *inferential semantics* can formally separate truth denotation and meaning. Moreover, this allows the definition of new complexity measures on formulas and proofs, that include both the syntactic and the semantic aspects.
It is worth noting that the role of Higher Order Logic in many application fields, most notably *computer science*, is important and significant: type systems for programming languages and models for the typed λ-calculus are relevant examples, e.g., typed λ-calculus [19] is a part of Church's formulation of Higher Order Logic [8]. Logic programming has also been deeply influenced by Higher Order Logic: the studies of Andrews [1] on higher order theorem provers are at the basis of the works of Miller on l-Prolog [22]. Thus, the *inferential semantics* proposed about some computational aspects of various formulations of HOL.
Further relevant developments can be envisaged when we restrict Higher Order Logic to some fragments of a logic programming nature, e.g. the higher order logic programming languages defined in [23], where the fundamental notion of uniform proof is introduced. Bai and Blair [4] and Wolfram [30] gave semantics for the classical higher order Horn clause fragment of λ-prolog. De Marco and Lipton [9] produced a model theory of resolution on HigherOrder Hereditary Harrop formulas (HOHH) with uniform proofs, through a constructive algebraic approach. The *inferential semantics* can be used to obtain, in the Higher Order Logic setting, something similar to the s-semantics approach studied by Levi's research group; it was used to have a notion of model meaningful from the computational point of view (see for example [6]). A preliminary study of extending the s-semantics approach to Higher Order Logic was presented in [21].
Moreover, Higher Order Logic is still a relevant topic also in fundamental research in logic: see for example the work of De Marco and Lipton on completeness and cut-elimination in the intuitionistic theory of types [10]. In such a perspective, one should note that the definition of the main tools of *inferential semantics* does not depend on the logic being classical: thus, they should extend reasonably to non-classical logics, and a completeness result for intuitionistic Higher Order Logic w.r.t. *inferential models* is work progress.

References

[1] P. B. Andrews. Resolution in Type Theory. *Journal of Symbolic Logic*, 36: 414–432, 1971.
[2] P. B. Andrews. General Models and Extensionality. *Journal of Symbolic Logic*, 37: 395–397, 1972.
[3] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: to Truth through Proof*. Academic Press, Orlando, 1986.
[4] M. Bai and H. Blair. General model theoretic semantics for higher'order Horn logic programming. *Logic Programming and Automated Reasoning, International Conference LPAR'92*, LNCS 624: 320–331, Springer 1992.
[5] J. Bell and M. Machover. *A Course in Mathematical Logic*. North-Holland, 1977.
[6] A. Bossi, M. Gabbrielli, G. Levi, and M. Martelli. The s-Semantics Approach: Theory and Applications. *Journal of Logic Programming*, 19/20: 149–197, 1994.

[7] S. Buss. First Order Proof Theory of Arithmetic. In S. Buss, editor, *Handbook of Proof Theory*, pages 79–147. Elsevier, Amsterdam, 1998.
[8] A. Church. A Formulation of Simple Theory of Types. *Journal of Symbolic Logic*, 5: 56–68, 1940.
[9] M. De Marco and J. Lipton. Uniform Algebras: a complete semantics for Higher Order Logic programming with HOHH Formulae, 2002. Preprint.
[10] M. De Marco and J. Lipton. Completeness and Cut-elimination in the Intuitionistic Theory of Types, *Journal of Logic and Computation*, 15: 821–854, 2005.
[11] P. Gentilini. Proof-Theoretic Modal PA-Completeness I: A System-Sequent Metric. *Studia Logica*, 63: 27–48, 1999.
[12] P. Gentilini. Proof-Theoretic Modal PA-Completeness II: The Syntactic Countermodel. *Studia Logica*, 63: 245–268, 1999.
[13] P. Gentilini. Proof-Theoretic Modal PA-Completeness III: The Syntactic Proof. *Studia Logica*, 63: 301–310, 1999.
[14] G. Gentzen. Untersuchungen uber das Logische Schliessen. In M. Szabo, editor, *Collected papers of G. Gentzen*. North Holland, 1969. Translated as 'Investigations into Logical Deduction'.
[15] J. Y. Girard. Linear Logic. *Theoretical Computer Science*, 50: 1–102, 1987.
[16] J. Y. Girard. *Proof Theory and Logical Complexity*. Bibliopolis, Napoli, 1987.
[17] L. Henkin. Completeness in the Theory of Types. *Journal of Symbolic Logic*, 15: 81–91, 1950.
[18] J. R. Hindley. *Basic Simple Type Theory*. Cambridge University Press, 1997.
[19] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and lambda-Calculus*. Cambridge University Press, 1986.
[20] M. Manzano. *Extensions of First Order Logic*. Cambridge University Press, 1996.
[21] M. Martelli, A. Messora, C. Palamidessi, and A. B. Hoa. Fixpoint Semantics of L-lambda. *Proc. GULPPRODE'94*, 1994. Peniscola (Spain).
[22] D. Miller. A Logic Programming Language with Lambda-Abstraction, Function Variables,and Simple Unification. In P. Schroeder-Heister, editor, *Extensions of Logic Programming*. Springer-Verlag, 1990.
[23] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform Proofs as a Foundation for Logic Programming. *Annals of Pure and Applied Logic*, 51: 125–157, 1991.
[24] J. C. Mitchell and E. Moggi. Kripke-Style Models for Typed Lambda Calculus. *Annals of Pure and Applied Logic*, 51:99–124, 1991.
[25] D. Prawitz. Hauptsatz for Higher order Logic. *Journal of Symbolic Logic*, 33:452–457, 1968.
[26] J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley, Reading Ma., 1967.
[27] G. Takeuti. *Proof Theory*. North Holland, second edition, 1987.
[28] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.
[29] J. Van Benthem and K. Doets. Higher-Order Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 1. Kluwer, 1983.
[30] D. Wolfram. A semantics for lambda-prolog. *Theoretical Computer Science*, 136: 277–289, 1994.