Algebraic and Topological Methods in Lambda Calculus

Antonino Salibra

University of Venice
salibra@dsi.unive.it

Abstract. The untyped lambda calculus was introduced around 1930 by Church as part of an investigation in the formal foundations of mathematics and logic. Although lambda calculus is a very basic language, it is sufficient to express all the computable functions. The process of application and evaluation reflects the computational behaviour of many modern functional programming languages, which explains the interest in the lambda calculus among computer scientists.
In this talk we give an outline of the theory of lambda abstraction algebras (LAA's). These algebras constitute an equational class and were introduced by Pigozzi and Salibra to provide an algebraic version of the untyped lambda calculus, in the same way cylindric and polyadic algebras algebraize the first-order predicate logic. Questions related to the functional representation of various subclasses of lambda abstraction algebras were investigated by Pigozzi and Salibra in a series of papers. The problem of representability for LAA's was solved by Goldblatt and Salibra by showing that every LAA is isomorphic to a representable one. Algebraic methods were also applied by Lusin and Salibra to study the structure of the lattice of lambda theories (i.e., extensions of lambda calculus that are closed under derivation) and by Salibra to study the structure of the lattice of the subvarieties of LAA. For every variety of LAA's there exists a lambda theory whose term algebra generates the variety. In particular, the variety LAA is generated by the term algebra of the minimal lambda theory. Finally, algebraic and topological methods were applied by Salibra to study the incompleteness of the semantics of lambda calculus.