Combinatory Logic and Program Synthesis

Jakob Rehof

Technical University of Dortmund, Germany
jakob.rehof@cs.tu-dortmund.de

Session: 6. Computational Logic

We survey a new approach to component-oriented program synthesis based on combinatory logic, which exploits the idea that (semi-) algorithms for solving the inhabitation problem in a combinatory logic can be used to generate programs from collections of components. Under the Curry-Howard isomorphism, combinatory type inhabitation is the question of provability for a corresponding propositional logic. We survey recent results, ongoing work, and open problems in combinatory logic synthesis.

For purposes of synthesis, the inhabitation problem must be considered *relative* to a given propositional theory (combinatory basis). The relativized inhabitation problem is undecidable already in simple types (Linial-Post theorem) which can consequently be seen as a Turing-complete logic programming language for synthesizing compositions. Under this viewpoint, combinatory type environments are programs and types are rules in such programs. In order to enhance the ability to express semantic specifications intersection types are introduced into composition synthesis, and we survey recent results on expressive power, algorithmics, and complexity of inhabitation in various fragments of combinatory logic with intersection types. Modal types are introduced as a foundation for meta-programming combinators. The resulting system supports multiple languages and logics in a highly flexible framework for composition synthesis. Based on a prototype implementation of the framework (CL)S, Combinatory Logic Synthesizer, we illustrate with practical examples as time permits. We will end the talk with a survey of open problems and future directions.