Development of Inductive Synthesis of Functional Programs

Fritz Wysotzki WYSOTZKI@CS.TU-BERLIN.DE
Technische Universität Berlin

Inductive program synthesis is yet in the state of fundamental research. The task consists of inductive construction of a program from pairs of given input-output examples or instantiated initial parts of the intended program using specific methods of generalisation. Well known is the field of Inductive Logic Programming (ILP) which is concerned with the induction of logical programs. The pioneering work in synthesis of functional programs was done by Summers 1977 using the programming language LISP. Summers approach was restricted to structural list problems i.e. his algorithms depend on the structure of a list but not of its content. After Summers other work on synthesis of functional programs was done using LISP, too. The main topic of this lecture will be the synthesis of functional programs in an abstract formalism, i.e. without using a special programming language. The basis are mainly theoretical investigations and programmed realisations of the algorithms which have been performed at the TU Berlin over many years. The lecture starts with some historical remarks, after that the theoretical basis is introduced. It is a term algebra including a special non-strict term (corresponding to the IF-THEN-ELSE) for realizing tests. A functional program is represented by a system of equations the left hand side of which are function variables (Ònames of subprogramsÓ) with parameters (Òformal parametersÓ), the right hand sides are terms, which may contain the function variables and there is a special term representing the Ómain programÓ. This system is called Recursive Program Scheme (RPS, Courcelle and Nivat 1978). It can be ÒsolvedÓ syntactically by an ordered sequence of finite terms (KLEENE-sequence), approximating the fixpoint solution which is an infinite term. This corresponds to unfolding the RPS.The induction principle works as follows: If one has a finite example term (i.e. with instantiated variables) one can try to Òexplain itÓ as being an element of a KLEENE-sequence of an RPS. Extrapolation with introduction of variables as generalisation principle and folding gives a hypothetical RPS which explains the given example term. The reliability of the hypothesis depends on the position of the given term in the KLEENE-sequence. One of the main problems which will be discussed is the detection and induction of subprograms in the example term by finding an appropriate segmentation. Another problem is how to get the example term. Two domain dependent approaches will be discussed: the integration of (mutually excluding) production rules and planning. In the latter case from a problem solving graph a shortest path tree (Universal Plan) is constructed. In the second step by a generalisation procedure similar to subsumption used in ILP a goal hierarchy is computed. This goal tree is then transformed into an initial program (program tree) which can be used as a basis for inductive construction of a RPS mentioned above. The main principles of our approach to the inductive construction of functional programs will be demonstrated by examples.