Abstract

Generating Equivalent Transformation Rules from Specifications of Problems

KIYOSHI AKAMA,† HIDEKATSU KOIKE† and EIICHI MIYAMOTO†

This paper proposes a new method for synthesis of programs from specifications, using a new computation model, called the "Equivalent Transformation model" (ET model). In the ET model, specifications are represented by a set of (extended) definite clauses, and the main part of the synthesis of programs is the generation of new equivalent transformation rules from the specification. In the ET model, programs may be improved by adoption of each new rule, and finally an efficient program may be obtained. The rule generation is different from the program transformation in Prolog, where a set of definite clauses (specification) is transformed into a new set of definite clauses (which is regarded as a program in logic programming).

In order to generate rules, a new concept of meta-descriptions is introduced, each of which represents (infinite) descriptions by instantiation. A meta-description is said to be transformed equivalently iff all the descriptions that are obtained from it by instantiation are transformed equivalently. An ET rule is obtained from each pair of meta-descriptions that are equivalent.

Meta descriptions are transformed in terms of meta-rules. Since we can use many meta-rules corresponding to unfolding, folding, and other PT rules, the proposed method can increase efficiency in computation at least as much as unfold/fold transformation in logic programming does. One of the most important characteristics of the proposed method is modularity. A meta-rule is modular in the sense that its correctness can be judged independently of other meta-rules. Meta-rules can be applied in any order, while program transformation rules for logic programs can not.

(Presented October 31, 1998)

† Department of System and Information Engineering, Hokkaido University