ON RELATIONSHIPS BETWEEN SEMANTIC OPERATORS IN LOGIC PROGRAMMING

MATTHIAS WENDT

Department of Computer Science
Dresden University of Technology
Dresden, Germany
mw177754@inf.tu-dresden.de

The introduction of negation in logic programming raises many questions, in particular how negation should to be interpreted semantically. Possible approaches involve classical negation as well as negation as failure – inspired by Prolog – or default negation – inspired by commonsense reasoning. These different ways of treating negation are captured by different semantic operators which can be associated with each given logic program. The operators provide a declarative semantics for logic programs in that the fixed points of the operators can be interpreted as intended (partial or total) models of the logic programs to which the operators are associated. Relationships between different operators, hence different semantics for logic programs, have been studied in the literature, but they are not yet fully understood.

Dung and Kanchanasut (1989), for example, proposed an operator, which was called the fixpoint completion of a program. This operator essentially performs an unfolding of the recursions through positive literals in the clauses of a program. It transforms a given logic program into another logic program, that has only negative literals in the bodies of its clauses. This is done in such a way that the treatment of negation as default negation in the original program corresponds to the treatment of negation as failure in the resulting program.

In this presentation, we are going to study relationships between semantic operators that are mediated by this fixpoint completion. These relationships originate in a similarity of behaviour of different semantic operators on logic programs resp. their fixpoint completion. In contrast to the work by Dung and Kanchanasut, we do not only relate the fixpoints of operators, which correspond to intended models, but whole (transfinite) sequences of powers of these operators.

We thus expose relationships not only between the mere consequence operators, but between concepts of dealing with negation in logic programs. This will serve as a step towards a deeper understanding of the relations and differences of the several approaches to the handling of negation and the several semantic operators connected to them.