Debugging in A-Prolog: A logical approach

Mauricio Osorio, Juan Antonio Navarro, José Arrazola

Universidad de las Américas, CENTIA
Sta. Catarina Mártir, Cholula, Puebla. 72820 México
{josorio, ma108907}@mail.udlap.mx

A-Prolog, Answer Set Programming or Stable Model Programming, is an important outcome of the theoretical work on Nonmonotonic Reasoning and AI applications of Logic Programming in the last 15 years. In the full version of this paper we study interesting applications of logic in the field of answer sets.

Two popular software implementations to compute answer sets, which are available and easy to find online, are DLV and SMODELS. Latest versions of these programs deal with disjunctive logic programs plus constraints. An important limitation is that, however, no tools for analyzing or debugging code have been provided. Sometimes, when computing models for a program, no answer sets are found while we were, in principle, expecting some of them to come out. We observed how an approach based on the three-valued logic G_3 can be useful to detect, for instance, constraints that are violated and invalidate all expected answer sets. These tools could help the user in finding the offending rules in the program and act in response correcting possible mistakes.

We introduce first the notion of *quantified knowledge* that is used to define an order among partial G_3 interpretations of logic programs. Then a notion of minimality between *implicitly-complete* interpretations, which can be uniquely extended to complete models, is defined in terms of this order. Such extended models are then called *minimal* models.

We defined then the *weak*-G_3 *semantics* as the set of minimal models for a given program, and the *strong*-G_3 *semantics* as the set of minimal models that are also definite (no atom is assigned to the undefined value of the G_3 logic). As a consequence of a characterization we provided for answer sets in terms of intermediate logics, we were able to prove that the strong-G_3 semantics corresponds exactly to the answer set semantics as defined for nested programs by Lifschitz, Tang and Turner [*Nested expressions in logic programs*, 1999].

The weak-G_3 semantics has, however, interesting properties we found useful for debugging purposes. Every consistent program has, for example, at least one minimal model and, since minimal models are not always definite, will detect atoms that are left indefinite. The intuition behind is that an answer set finder cannot decide, for these indefinite atoms, if they are either true or false and thus rejecting a possible model. It is also discussed how this ideas can be applied, using a simple transformation of constraints into normal rules, to detect violated constraints in programs.

Acknowledgements This research is sponsored by the Mexican National Council of Science and Technology, CONACyT (project 35804-A). Full version of this paper available online at http://www.udlap.mx/~ma108907/iclp.