Answer Set Solvers and the Basic Concepts of Answer Sets

Marvin Beese, mabeese@uni-potsdam.de, 786300

University of Potsdam

Abstract. Answer Set Solvers have close relations to other programming paradigms but also bring their own advantages. With Answer Set Programming problems can be solved under consideration of constraints and basic expression concepts. This is a brief introduction to ASP.

1 Introduction

This summary is based on 'Answer Sets and the Language of Answer Set Programming' and 'The Answer Set Programming Paradigm' by Vladimir Lifschitz and Tomi Janhunen, Ilkka Niemelä and deals with the basic concepts of Answer Set Solving. Answer Set Solving is used for fast and reliable problem solving of constraint satisfaction problems. Its main field of application is Artificial Intelligence in Computer Science. First I am going to describe the relations to other neighboring fields and examine deeper how an Answer Set Solver works. First I will examine its grounding and solving phase and describe later its basic programming expressions.

2 Answer Set Solvers

Answer Set Programming (ASP) is a declarative way to solve searching problems. A Program consists of a set of statements that describe a problem whose solution are stable models/answer sets. ASP has close relations to constraint programming, propositional satisfiability, linear and integer programming, logic programming and Prolog. Where Prolog is mainly engineered for finding proofs or answer substitutions to a problem instance, ASP finds solutions to a problem instance, is fully declarative and therefore neither the order of the rules nor the order of the literals matter. Problems are represented as constraints, whose complete solutions can be found by assigning variables in a satisfying way. In ASP problems have a structured representation where a problem statement can be developed independently of data, moreover smaller constraints can be used effectively to model more difficult constraints, which enables an incremental development of applications. ASP problems are modeled using facts and rules. Facts are either given or are generated through rules. Problem Solving includes two steps. The first step, grounding, is used to evaluate more complicated data structures and to instantiate variables. Solving is the second step, where the search of answer sets takes place and where advanced search methods are used.

3 Basic Concepts of Answer Sets

A positive program, is a set of rules of the form $A_0 : -A_1, ..., A_n$. This program doesn't contain any form of negation. In the grounding phase, variables will be substituted for specific values and the result will be sets of facts/ground atoms. If a program does contain negation, grounding gives a set of rules of the form $A_0 : -A_1, ..., A_m$, not $A_m + 1, ...,$ not A_n, where negation is included. Negation as failure exists in programs of the form

$$p(1).\ p(2).\ p(3).\ q(2).\ q(3).\ q(4).r(X) : -p(X),\ not\ q(X). \qquad (1)$$

This means, that any attempt to generate $q(X)$ using the rules of the program, including the rule $r(X)$ itself, would fail. In that case, the answer set would look like $p(1)$ $p(2)$ $p(3)$ $q(2)$ $q(3)$ $q(4)$ $r(1)$. To decide, whether a set of ground atoms is an answer set, on has to form the reduct. Some rules are dropped, and some negated atoms are dropped. As the negated atoms are dropped only the positive parts will be included.
In ASP arithmetic operations and comparisons can be found in rules

$$r(X + Y) :-\ \ p(X), q(Y), X < Y \qquad (2)$$

as well as disjunctive rules. In disjunctive rules the head of the rule can have disjunction on several atoms. Choice rules include a list of atoms in the head of the rule in curly braces, where atoms are choosen in all possible ways. Bounds can be specified for both the lower bound as well as for the upper bound through numbers on the before left and after right side of the braces. A constraint rule is a disjunctive rule with 0 constraints in the head of the rule, this eliminates all answer sets that satisfy the body. Classical negation can be used for representing incomplete information. For example: $p(a), p(b), -p(c)$ means that a and b have property p, but c does not. Beyond basic answer set programming, to select the best solution among alternatives, optimization comes in handy. For example with the expression $\#minimize\{w_1, 1 : atom_1; ...; w_n, n : atom_n\}$. weights w_i are assigned to atoms $atom_i$. The goal of minimization is to minimize the sum of weights w_i to find the optimal solution.

4 Conclusion

With ASP problems can be modeled fast and independently of data input, which leads to elaboration tolerance. Answer Set Solvers can solve programs described by constraints very fast. This gives an opportunity for modern problem solving in the research and industrial field of artificial intelligence.

References

1. Vladimir Lifschitz: Answer Sets and the Language of Answer Set Programming. AI Magazine 37(3): 7-12 (2016)
2. Tomi Janhunen, Ilkka Niemelä: The Answer Set Programming Paradigm. AI Magazine 37(3): 13-24 (2016)