Abstract. The goal of this thesis is to extend Answer Set Programming (ASP) with declarative heuristics, preferences, and online planning capabilities. For declarative heuristics, the thesis presents a general declarative approach for incorporating domain-specific heuristics into ASP solving by means of logic programming rules. For preferences, the approach developed in my thesis and the resulting asprin system provide a general and flexible framework for quantitative and qualitative preferences in ASP. For online planning, the goal of my thesis is to integrate different approaches to online planning with incomplete information in a unified ASP approach.

1 Introduction

Answer Set Programming (ASP; [1]) is a well established approach to declarative problem solving. Rather than solving a problem by telling a computer how to solve the problem, the idea is to simply describe what the problem is and leave its solution to the computer. The success of ASP is due to the combination of a rich yet simple modeling language with high-performance solving capacities. Modeling has its roots in the fields of Knowledge Representation and Logic Programming, while solving is based in methods from Deductive Databases and Satisfiability Testing (SAT; [2]). ASP programs resemble Prolog programs, but they are interpreted according to the stable models semantics [12], and the underlying solving techniques are closely related to those of modern SAT solvers. The goal of my doctorate is to develop different extensions of ASP: for declarative heuristics, preferences, and online planning.

2 Declarative Heuristics

From the solving perspective, for solving real-world problems in ASP, it is sometimes advantageous to take an application-oriented approach by including domain-specific information. On the one hand, domain-specific knowledge can be added for improving deterministic assignments through propagation. On the other hand, domain-specific heuristics can be used for making better nondeterministic assignments. To this end, in my thesis I introduce a general declarative framework for incorporating domain-specific heuristics into ASP solving.
Extensions of ASP: Heuristics, Preferences and Online Planning

[11], using a directive \texttt{#heuristic} whose arguments allow us to express various modifications to the solver’s heuristic treatment of atoms. The directives are interpreted as a new type of rules, that are subsequently exploited by the solver when it comes to choosing an atom for a non-deterministic truth assignment. The heuristic framework offers completely new possibilities of applying, experimenting, and studying domain-specific heuristics in a uniform setting. In the current stage, heuristic directives are an integral part of \texttt{clingo5} and have already been used in some real world applications. The next step is to extend the approach using machine learning techniques for automatically learning heuristic rules.

3 Preferences

Another extension that is often necessary in real-world applications is being able to represent and reason about preferences. This was realized quite early in ASP, leading to many approaches to preferences [7, 4, 14]. Departing from there, the approach developed in my thesis [5, 6] and the resulting \texttt{asprin}\textsuperscript{1} system provide a general and flexible framework for quantitative and qualitative preferences in ASP. This framework is \textit{general} and captures many of the existing approaches to preferences. It is \textit{flexible}, providing means for the combination of different types of preferences. And it is also \textit{extensible}, allowing for an easy implementation of new approaches to preferences. The next steps for this part of my thesis are to finish the implementation of a stable and user-friendly version of the system, to implement new types of preferences in the system (i.e., CP-nets), and to integrate unsatisfiable-core solving techniques into the approach.

4 Online Planning

The third part of my thesis is focused on extensions for online planning with ASP. Planning is one of the earlier applications of ASP [13]. It has been extended to deal with incomplete information about the initial state and/or sensing actions, and for solving conformant [10, 15, 17] and conditional planning [16] problems. Alternative approaches outside ASP for dealing with incomplete information and sensing actions include \textit{assumption-based planning} [8] and \textit{continual planning} [3]. The goal of this last part of the thesis is to integrate these techniques in a unified ASP framework, and apply it to a real-world application in robotics.

References


\textsuperscript{1} \texttt{asprin} stands for “ASP for preference handling”.


