

# Abstract

Answer Set Programming (ASP) is a purely declarative programming paradigm based on nonmonotonic reasoning and logic programming. The idea of ASP is to represent a given computational problem by a logic program such that its answer sets correspond to solutions, and then, use an answer set solver to find such solutions. The ASP language is very expressive and allows the representation of high-complexity problems (i.e., every problem in the second level of the polynomial hierarchy); unfortunately, the expressive power of the language comes at the price of an elevated cost of answer set computation. Even if this fact has initially discouraged the development of ASP system, nowadays several implementations are available, and the interest for ASP is growing in the scientific community as well as in the field of industry.

In the last few years, significant technological enhancements have been achieved in the design of computer architectures, with a move towards the adoption of multi-core microprocessors. As a consequence, Symmetric Multi-Processing (SMP) has become available even on non-dedicated machines. Indeed, at the time of writing, the majority of computer systems and even laptops are equipped with (at least one) dual-core processor. However, in the ASP context, the available systems were not designed to exploit parallel hardware; thus significant improvements can be obtained by developing ASP evaluation techniques that allow the full exploitation of the computational resources offered by modern hardware architectures.

The evaluation of ASP programs is traditionally carried out in two steps. In the first step an input program  $\mathcal{P}$  undergoes the so-called instantiation process, which produces a program  $\mathcal{P}'$  semantically equivalent to  $\mathcal{P}$  but not containing any variable; in turn,  $\mathcal{P}'$  is evaluated by using a backtracking search algorithm in the second step.

The aim of this thesis is the design and the assessment of a number of parallel techniques devised for both steps of the evaluation of ASP programs. In particular, a three-level parallel instantiation technique is presented for improving the efficiency of the instantiation process which might become a bottleneck in common situations, especially when huge input data has to be dealt with. Moreover, a parallel multi-heuristics search algorithm and a parallel lookahead technique have been conceived for optimizing the second phase of the evaluation.

The mentioned parallel techniques has been implemented in the state-of-the-art ASP system DLV. An extensive experimental analysis has been carried out in order to assess the performance of the implemented prototypes. Experimental results have confirmed the efficiency of the implementation and the effectiveness of those techniques.