Abstract Solvers for Answer Set Programming and Beyond

Marco Maratea

University of Genoa, Italy

Abstract. Abstract solvers are a tool for describing solving algorithms via graphs, that entail some advantages w.r.t. traditional ways of describing solving algorithms, e.g. pseudo-code. In the letter, first I briefly mention the main ideas and advantages of abstract solvers. Then, I give useful pointers to the usage of abstract solvers in Answer Set Programming, and beyond.

Abstract solvers are a relatively new tool for describing solving algorithms for given reasoning tasks. Usually, solving algorithms are presented by means of pseudo-code descriptions, but some communities have experienced that analyzing algorithms on this basis may not be fruitful. Thus, more formal descriptions which allow, e.g. for a uniform representation, have been studied and developed. In abstract solvers, such desiderata are met by relying on graphs: one characterizes the states of computation as nodes of a graph, the techniques as arcs between nodes, the whole solving process as a path in the graph, and formal properties of the algorithms are reduced to related graph's properties. Thus, abstract solvers (i) allow for a more deep understanding of the ideas behind a solving procedure, (ii) simplify stating and proving formal properties of the related algorithms, e.g. correctness, and (iii) support the comparison of different search procedures by means of comparing their related graphs.

The use of abstract solvers was initiated by Nieuwenhuis et al. in [14]. In this work the authors first presented an abstract solver related to the DPLL procedure for SAT. Then, they presented two extensions to this graph: (*i*) a first extended graph to describe CDCL SAT solving, i.e. involving backjumping and learning techniques, by means of modular addition of transition rules, and (*ii*) a second graph to solve Satisfiability Modulo Theories (SMT) problems with certain logics via a lazy approach [15], i.e. where the SAT calls are made to provide satisfying assignments of the Boolean abstraction of the SMT problem that are then checked for "SMT consistency". Lierler [6] then imported this methodology in ASP, by first designing abstract solvers for some backtracking-based ASP solvers for non-disjunctive ASP solving, then enhanced to include backjumping and learning techniques [7]. Another extension for describing CASP solvers, i.e. systems able to deal with a combination of ASP and constrain programming, a language useful to represent and reasoning on hybrid domains, has been put forward in [8]. Other papers on abstract solvers are [9], where solvers for different formalisms, i.e. ASP and SAT with Inductive Definitions, are compared

by means of comparison of the related graphs. Brochenin et al. [4] exploited abstract solvers for cautious reasoning in ASP.

Abstract solvers for some disjunctive answer set solvers implementing basic backtracking have been introduced by Brochenin et al. [1], and then showed, by the same authors, to be instantiations of a more general graph template that includes the techniques implemented by these solvers in [2].

Application of abstract solvers outside ASP include Quantified Boolean Formulas [5] and certain reasoning tasks of Dung's Argumentation Frameworks [3]. Moreover, starting from a developed concept of modularity in answer set solving [10], abstract modeling of solvers for multi-logic systems are presented in [11–13].

References

1. Brochenin, R., Lierler, Y., Maratea, M.: Abstract disjunctive answer set solvers. In: Schaub, T., Friedrich, G., O'Sullivan, B. (eds.) Proc. of the 21st European Conference on Artificial Intelligence, ECAI 2014. FAIA, vol. 263, pp. 165–170. IOS Press (2014)
2. Brochenin, R., Lierler, Y., Maratea, M.: Abstract disjunctive answer set solvers via templates. In: Theory and Practice of Logic Programming. To appear (2016)
3. Brochenin, R., Linsbichler, T., Maratea, M., Wallner, J.P., Woltran, S.: Abstract solvers for dung's argumentation frameworks. In: Black, E., Modgil, S., Oren, N. (eds.) Theory and Applications of Formal Argumentation - 3rd International Workshop, TAFA 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9524, pp. 40–58. Springer (2015)
4. Brochenin, R., Maratea, M.: Abstract answer set solvers for cautious reasoning. In: Vos, M.D., Eiter, T., Lierler, Y., Toni, F. (eds.) Proc. of the Technical Communications of the 31st International Conference on Logic Programming (ICLP 2015). CEUR Workshop Proceedings, vol. 1433. CEUR-WS.org (2015)
5. Brochenin, R., Maratea, M.: Abstract solvers for quantified boolean formulas and their applications. In: Gavanelli, M., Lamma, E., Riguzzi, F. (eds.) Advances in Artificial Intelligence - Proc. of the XIVth International Conference of the Italian Association for Artificial Intelligence (AI*IA 2015). Lecture Notes in Computer Science, vol. 9336, pp. 205–217. Springer (2015)
6. Lierler, Y.: Abstract answer set solvers. In: de la Banda, M.G., Pontelli, E. (eds.) Proc. of the 24th International Conference on Logic Programming (ICLP 2008). vol. 5366, pp. 377–391. Springer (2008)
7. Lierler, Y.: Abstract answer set solvers with backjumping and learning. Theory and Practice of Logic Programming 11(2-3), 135–169 (2011)
8. Lierler, Y.: Relating constraint answer set programming languages and algorithms. Artificial Intelligence 207, 1–22 (2014)
9. Lierler, Y., Truszczynski, M.: Transition systems for model generators - A unifying approach. Theory and Practice on Logic Programming 11(4-5), 629–646 (2011)
10. Lierler, Y., Truszczynski, M.: Modular answer set solving. In: Late-Breaking Developments in the Field of Artificial Intelligence. AAAI Workshops, vol. WS-13-17. AAAI (2013)

11. Lierler, Y., Truszczynski, M.: Abstract modular inference systems and solvers. In: Flatt, M., Guo, H. (eds.) Proc. of the 16th International Symposium on Practical Aspects of Declarative Languages (PADL 2014). Lecture Notes in Computer Science, vol. 8324, pp. 49–64. Springer (2014)
12. Lierler, Y., Truszczynski, M.: An abstract view on modularity in knowledge representation. In: Bonet, B., Koenig, S. (eds.) Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015). pp. 1532–1538. AAAI Press (2015)
13. Lierler, Y., Truszczynski, M.: On abstract modular inference systems and solvers. Artificial Intelligence 236, 65–89 (2016)
14. Nieuwenhuis, R., Oliveras, A., Tinelli, C.: Solving SAT and SAT modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). Journal of the ACM 53(6), 937–977 (2006)
15. Sebastiani, R.: Lazy satisability modulo theories. Journal of Satisfiability, Boolean Modeling and Computation 3(3-4), 141–224 (2007)