

Answer Set Solving in Practice

A Tutorial at AAAI'13

Martin Gebser and Torsten Schaub
University of Potsdam, Germany

<http://www.cs.uni-potsdam.de/~torsten/aaai13tutorial/summary>

Short Description

This half-day tutorial presents a practical introduction to Answer Set Programming (ASP), aiming at using ASP languages and systems for solving application problems. Starting from the essential formal foundations, it introduces ASP's solving technology, modeling language and methodology, and systems.

What makes this tutorial different is its focus on putting ASP at work. This comprises a good understanding of ASP solving technology and systems as well as basic skills in ASP's modeling capacities.

Description

Answer Set Programming (ASP) is a declarative problem solving approach, combining a rich yet simple modeling language with high-performance solving capacities. ASP is particularly suited for modeling problems in the area of Knowledge Representation and Reasoning involving incomplete, inconsistent, and changing information. From a formal perspective, ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way (being more compact than SAT). Applications of ASP include automatic synthesis of multiprocessor systems, decision support systems for NASA shuttle controllers, reasoning tools in systems biology, and many more. The versatility of ASP is also reflected by the ASP solver `clasp`, developed at the University of Potsdam, and winning first places at international ASP, CASC, MISC, PB, and SAT competitions.

The tutorial aims at acquainting the participant with ASP's modeling and solving methodology, enabling her/him to independent problem solving using ASP systems. To this end, the tutorial starts with an introduction to the essential formal concepts of ASP, needed for understanding its semantics and solving technology. In fact, ASP solving rests on two major components: A grounder turning specifications in ASP's modeling language into propositional logic programs and a solver computing a requested number of answer sets of the given program. Building on the aforementioned formal concepts, we provide a characterization of ASP's inference schemes that are in turn mapped into algorithms relying on advanced Boolean Constraint technology. Similarly, ASP's grounding inferences are discussed in conjunction with (deductive) database techniques. The remainder of the tutorial is dedicated to applying ASP, involving an introduction to ASP's modeling language, its solving methodology, and advanced techniques fostering scalability.

All involved ASP systems are freely available from <http://potassco.sourceforge.net>.

Outline of the Tutorial

- Motivation
- Introduction
- Basic modeling
- Grounding
- Solving
- Systems
- Advanced modeling

Potential Target Audience

- Graduate students in AI
- Teachers of AI
- Researchers and industrials addressing combinatorial problems

Interest to AAAI Audience

ASP provides a declarative tool for modeling various problems typical to Knowledge Representation and Reasoning in particular and AI in general. The unique pairing of declarativeness and performance allows for concentrating on an actual problem, rather than a smart way of implementing it. The ASP approach is not only highly suitable for the practitioner solving an AI problem at hand but also for disseminating many basic AI techniques through teaching their (executable) formalization in ASP.

Material

- Lecture slides
- Lecture examples
- Potassco, the Potsdam Answer Set Solving Collection
 - ASP Teaching material
 - ASP Book
 - User’s manual
 - Overview article
- This file as PDF or HTML

Resumes

Martin Gebser and Torsten Schaub are working on ASP in the Knowledge Representation and Reasoning group at the University of Potsdam. The group’s activity in ASP has led to the open source project *Potassco* [4], the *Potsdam Answer Set Solving Collection*, bundling tools for ASP developed at Potsdam, and hosted at potassco.sourceforge.net. Potassco comprises more than a dozen ASP-related systems, among them the award winning solver *clasp*.

Contact

Drop us an email at {gebser, torsten}@cs.uni-potsdam.de for any questions.

References

- [1] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003.
- [2] G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.
- [3] T. Eiter, G. Ianni, and T. Krennwallner. Answer Set Programming: A Primer. In S. Tessaris, E. Franconi, T. Eiter, C. Gutierrez, S. Handschuh, M. Rousset, and R. Schmidt, editors, *Fifth International Reasoning Web Summer School (RW’09)*, volume 5689 of *Lecture Notes in Computer Science*, pages 40–110. Springer-Verlag, 2009.
- [4] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

- [5] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.
- [6] M. Gebser, B. Kaufmann, and T. Schaub. Conflict-driven answer set solving: From theory to practice. *Artificial Intelligence*, 187-188:52–89, 2012.
- [7] M. Gelfond. Answer sets. In V. Lifschitz, F. van Harmelen, and B. Porter, editors, *Handbook of Knowledge Representation*, chapter 7, pages 285–316. Elsevier Science, 2008.
- [8] V. Lifschitz. Foundations of logic programming. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–127. CSLI Publications, 1996.