



## Special Issue on Answer Set Programming

Torsten Schaub<sup>1</sup> · Stefan Woltran<sup>2</sup>

© Springer-Verlag GmbH Germany, part of Springer Nature 2018

“What is Answer Set Programming? And what can it do for me?” might have been the questions you had in mind when taking up this special issue. Let’s start with your first question. Answer Set Programming, or ASP for short, is a rule-based formalism for modeling and solving knowledge-intensive combinatorial (optimization) problems. What makes ASP attractive is its combination of a declarative modeling language with highly effective solving engines. This allows you to concentrate on specifying—rather than programming the algorithm for solving—a problem at hand. Historically, ASP has its roots in deductive databases, logic programming, and non-monotonic reasoning; its solving engines draw on the same technology as solvers for satisfiability testing. Given this origin, ASP is tailored to support closed as well as open world reasoning, which makes it predestined for knowledge representation and reasoning tasks.

The answer to your second question boils more or less down to the availability of off-the-shelf tools for knowledge representation and reasoning. You may use them to solve your own combinatorial (optimization) problems or to teach various forms of reasoning in your courses. What appeals to many users is the ease of modeling and the resulting transparency and flexibility of the problem encoding. Given that (most) such encodings are then effectively solved by ASP systems is yet another asset. Finally, it is interesting to observe that computer scientists often have a harder time with ASP than laymen, simply because they are not used to leave the control of their program to a computer.

And by now the question “What’s the catch?” may have formed in your mind. From an academic perspective, ASP technology is rather robust and mature. The balancing act stems from the fact that it is still an infant when it comes to

industrial use. Indeed, ASP is fully declarative in decoupling the specification of a problem from the execution of the underlying solving algorithm. Nonetheless, two equivalent declarative specifications may yet lead to a quite different solving performance. This is actually where still a blank field of research lies, namely, in automatic source code optimization. Another such open field is broadly speaking software engineering, where new ideas are needed given that ASP fully abstracts from an execution model. This makes, for instance, most traditional debugging techniques, like tracing, inapplicable.

So, what’s in the box? Our special issue aims at giving you an overview on what ASP is about and where it can be applied. Apart from our personal review of ASP and its current role, we gathered a series of technical papers: The issue starts with an article introducing the logical foundations of ASP, which are quite different from the original fixpoint characterization. The next article is dedicated to a particular feature of ASP’s modeling language, namely, aggregates. After these ASP-centered contributions, the next paper elaborates upon the relation to neighboring solving paradigms. The following bulk of articles is devoted to the application of ASP in various domains, viz. agent programming, robotics, linguistics, and bioinformatics. The final contribution surveys the industrial impact of ASP.

This technical row of papers is complemented by interviews with two contemporary witnesses of ASP’s development over the last decades, namely, Vladimir Lifschitz and Gerhard Brewka. Talking to Vladimir as the co-inventor of ASP provided us with first hand insights into the motivations and interesting peripheral phenomena during the creation of ASP. Similarly, Gerd was at the center of the non-monotonic reasoning community in Europe and fostered its development at various research locations.

Finally, we solicited system and project articles via an open call in order to reflect ongoing activities in the ASP community. These short papers range from system descriptions of state-of-the-art ASP systems over certain extensions of the ASP language to further application domains where ASP proved to be successful.

---

✉ Torsten Schaub  
torsten@cs.uni-potsdam.de  
Stefan Woltran  
woltran@dbai.tuwien.ac.at

<sup>1</sup> University of Potsdam, Potsdam, Germany

<sup>2</sup> TU Wien, Vienna, Austria

It comes without saying that this issue had been impossible without the great support of many people. Foremost, we would like to thank all authors for their dedication to promoting ASP through their articles. Given that all papers were examined by two reviewers, we would like to extend our gratitude to Daniela Incezan, Jacques Nicolas, Michael Morak, Orkunt Sabuncu, Rolf Schwitter, and Yulyia Lierler, who helped us with reviewing (in addition to the authors). We are particularly grateful to Anni-Yasmin Turhan and Vidhya Velayudhan and her team for their encouragement and guidance through the whole process. And of course a big thanks goes to all members of our research groups for their daily support!

Last but not least, let us mention that we conceived this special issue to be complementary to the one of the AI Magazine which appeared in fall 2016. You may find it interesting to have a look at both special issues on ASP. And now enjoy!

## 1 Content

### 1.1 Technical contributions

- Answer Set Programming Unleashed!  
Torsten Schaub and Stefan Woltran
- Answer Set Programmign from a Logical Point of View  
Pedro Cabalar, David Pearce, and Agustín Valverde Ramos
- Aggregates in Answer Set Programming  
Mario Alviano and Wolfgang Faber
- Answer Set Programming Related with Other Solving Paradigms  
Tomi Janhunnen
- Answer Set Planning in Single- and Multi-Agent Environments  
Marcello Balduccini and Tran Cao Son
- Applications of ASP in Robotics  
Esra Erdem and Volkan Patoglu
- Answer Set Programming in Linguistics  
Peter Schüller
- ASP Applications in Bio-informatics: A short tour  
Alessandro Dal Palú, Agostino Dovier, Andrea Formisano, and Enrico Pontelli
- Industrial Applications of Answer Set Programming  
Andreas Falkner, Gerhard Friedrich, Konstantin Schekotihin, Richard Taupe, and Erich Christian Teppan

### 1.2 System descriptions

- The ASP System DLV: Advancements and Applications  
Weronika Adrian, Mario Alviano, Francesco Calimeri, Bernardo Cuteri, Carmine Dodaro, Wolfgang Faber, Davide Fuscá, Nicola Leone, Marco Manna,

Simona Perri, Francesco Ricca, Pierfrancesco Veltri, and Jessica Zangari

- The Potsdam Answer Set Solving Collection 5.0  
Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Patrick Lühne, Philipp Obermeier, Max Ostrowski, Javier Romero, Torsten Schaub, Sebastian Schellhorn, and Philipp Wanko
- Cross-Translating Answer Set Programs Using the ASPTOOLS Collection  
Tomi Janhunnen
- Developing ASP programs with ASPIDE and LoIDE  
Francesco Calimeri, Stefano Germano, Eliana Palermi, Kristian Reale, and Francesco Ricca
- The DLVHEX System  
Thomas Eiter, Stefano Germano, Giovambattista Ianni, Tobias Kaminski, Christoph Redl, Peter Schüller, and Antonius Weinzierl
- Dynamic Programming on Tree Decompositions with D-FLAT  
Michael Abseher, Bernhard Bliem, Markus Hecher, Marius Moldovan, and Stefan Woltran
- Stream Reasoning with LARS  
Harald Beck, Minh Dao-Tran, Thomas Eiter, and Christian Folie

### 1.3 Project reports

- Lazy Grounding for Dynamic Configuration  
Thomas Eiter, Gerhard Friedrich, Richard Taupe, and Antonius Weinzierl
- Advanced Solving Technology for Dynamic and Reactive Applications  
Gerhard Brewka, Stefan Ellmauthaler, Gabriele Kern-Isberner, Philipp Obermeier, Max Ostrowski, Javier Romero, Torsten Schaub, and Steffen Schieweck
- ASP for Conformance Analysis and Explanation of Clinical Guidelines Execution  
Luca Piovesan, Matteo Spiotta, Paolo Terenziani, and Daniele Theseider Dupré
- Design Space Exploration with ASP  
Christian Haubelt, Kai Neubauer, Torsten Schaub, and Philipp Wanko
- ASP applied to Coreference Resolution and Semantic Similarity  
Peter Schüller
- ArgueApply: Abstract Argumentation at Your Fingertips  
Jörg Pührer

### 1.4 Interviews

- Vladimir Lifschitz
- Gerd Brewka

## 2 Service

### 2.1 Conferences and workshops

- International Joint Conference on Artificial Intelligence (IJCAI)
- European Conference on Artificial Intelligence (ECAI)
- AAAI Conference on Artificial Intelligence (AAAI)
- International Conference on the Principles of Knowledge Representation and Reasoning (KR)
- International Conference on Logic Programming (ICLP)
- International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)

- European Conference on Logics in Artificial Intelligence (JELIA)

### 2.2 Journals

- Artificial Intelligence
- Theory and Practice of Logic Programming
- ACM Transactions on Computational Logic