What's hot in the Answer Set Programming Competition

Marting Gebser
University of Potsdam, Germany
gebser@cs.uni-potsdam.de

Marco Maratea
Università di Genova, Italy
marco@dibris.unige.it

Francesco Ricca
Università della Calabria, Italy
ricca@mat.unical.it

Abstract

Answer Set Programming (ASP) is a declarative programming paradigm with roots in logic programming, knowledge representation, and non-monotonic reasoning. The ASP competition series aims at assessing and promoting the evolution of ASP systems and applications. Its growing range of challenging application-oriented benchmarks inspires and showcases continuous advancements of the state of the art in ASP.

Introduction

Answer Set Programming (ASP) (Brewka, Eiter, and Truszczyński 2011) is a declarative programming approach to knowledge representation and reasoning, having close relationships to neighboring areas like Boolean Satisfiability and Constraint Programming. Since the first edition in 2007, the ASP competition series assesses ASP systems on challenging benchmarks in order to promote the state of the art (Calimeri et al. 2012).

The sixth edition (Gebser, Maratea, and Ricca 2015) of the ASP competition took place in affiliation with the 13th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR 2015).[1] Similar to past events (Calimeri et al. 2015), it featured several tracks based on language features of benchmark encodings in ASP-Core-2 format[2] as well as on the number of processors allocated to participant systems.

In the following, we spotlight hot aspects of the sixth edition of the ASP competition, including its benchmark selection process, the advancements of participant systems, a newly introduced Marathon track, and an on-site modeling event complementing the "in silico" system competition.

Benchmark Suite and Selection

The competition benchmarks included 26 domains for which encodings and instances were already available from earlier editions (Calimeri, Ianni, and Ricca 2014). For four of these domains, the benchmark authors kindly provided fresh and more challenging instance sets, suitable for making meaningful system comparisons. Application-oriented benchmarks from six new domains were submitted in addition, thus doubling the number of domains stemming from applications. Overall, this amounts to 32 benchmark domains, comprising 5058 instances in total, modeling challenging problems from various areas including: artificial intelligence, databases, games, graph theory, industrial configuration, planning and scheduling, system synthesis, etc.

While instances had been drawn purely at random in the past (in 2013 also taking hardness estimates by benchmark authors into account), the sixth edition of the ASP competition aimed at balancing the hardness of instances selected per benchmark domain. To this end, an instance selection process inspired by the 2014 SAT Competition[3] has been incorporated. First, the empirical hardness of all available instances has been evaluated by running the three top-performing systems from the previous edition, and then a balanced selection was made among instances of varying difficulty. Running the three reference systems exhaustively took about 212 CPU days on the competition platform and led to a classification of the available instances. Instances solved in less than 20 seconds by each of the three systems as well as those where all of them failed in the grounding phase were classified as too easy or non-groundable. While such instances remain uninformative regarding a system comparison and were thus dropped, the other instances are partitioned into four hardness categories. The instance selection was then balanced by aiming at 20% of instances from each category plus another 20% picked freely, where concrete instances are drawn at random among the respective candidates. Although a perfect balancing was often impossible due to underpopulated hardness categories, an approximate compensation was feasible for all but four domains (with easy instances only). These domains were discarded, so that benchmarks from 28 domains, 12 of which are inspired by applications, were eventually included in the competition.

Notably, the instance selection process has itself been implemented in ASP, based on a factual representation of runtime data. As a result, the material collected on the website[4] of the sixth edition of the ASP competition contributes a

[1] http://lpnmr2015.mat.unical.it/

[2] https://www.mat.unical.it/aspcomp2013/ASPStandardization/

[3] http://www.satcompetition.org/2014/

[4] http://aspcomp2015.dibris.unige.it/

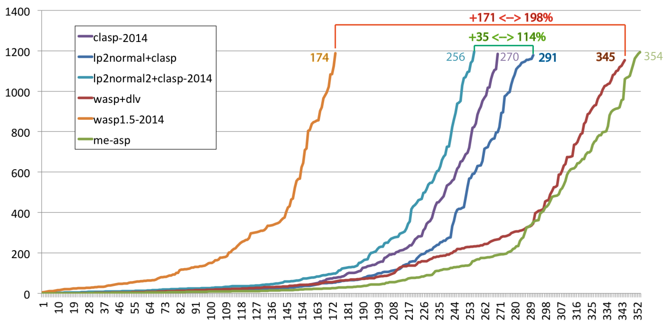

Figure 1: Progress of participant systems from 2014 to 2015.

broad range of benchmark domains, empirical hardness data for all instances provided by benchmark authors, as well as a so-called uniform encoding of balanced instance selection. While 20 instances per domain were picked for the competition, the selection process can be adopted and customized to furnish other representative benchmark suites in the future.

Advancement of the State of the Art

The competition featured 13 systems, submitted by three teams. We here focus on single-processor systems, forming the majority of 11 submissions. Full details are provided on the website[4] of the sixth edition of the ASP competition.

In the Main track, each system was allotted 20 minutes per run, including the time for grounding, preprocessing, and search. While the scoring system distinguishes between decision/query and optimization problems, up to 5 points could be earned per instance, leading to a perfect score of 2800 over 560 instances in total. The first three places were taken by the top-performing systems from each team: the multi-engine system ME-ASP (Maratea, Pulina, and Ricca 2014; 2015) with a score of 1971, the combined system WASP+DLV (Alviano, Dodaro, and Ricca 2014; Alviano et al. 2015; Leone et al. 2006) with score 1938, and LP2NORMAL+CLASP integrating dedicated preprocessing (Bomanson, Gebser, and Janhunen 2014; Bomanson and Janhunen 2013) and search (Gebser et al. 2015) with score 1760.

The advancements relative to reference systems from the previous edition are visualized in Figure 1, plotting numbers of solved instances indicated on the x-axis within runtimes given on the y-axis. While the winner system from 2014, labeled CLASP-2014, could solve 270 instances out of the updated benchmark set, LP2NORMAL+CLASP, WASP+DLV, and ME-ASP were able to solve 21, 75, or 84 instances more, respectively. When comparing LP2NORMAL+CLASP and WASP+DLV to earlier versions from 2014, labeled LP2NORMAL2+CLASP-2014 or WASP1.5-2014, improvements of the current systems amount to 35 or 171 more solved instances, respectively.

The additional margin of the new entrant ME-ASP, a portfolio solver that (roughly) combines the participant systems from 2014, clearly shows the benefit of integrating diverse approaches along with well-configured algorithm selection for tackling the variety of competition benchmarks. In summary, the performance results exhibit significant advancements and growing maturity of ASP systems, despite of only one year development time since the previous edition.

Marathon Track

As an idea borrowed from past QBF evaluations, the three top-performing systems were granted an order of magnitude more time per instance, i.e., 3 hours rather than 20 minutes only, in the Marathon track. This shifts the focus to the performance on hard instances, especially those that could not be solved by any system in the Main track.

Accumulating the results of all three systems, a significant portion of 11% more instances than in the Main track could be solved. Interestingly, the first and the second place are swapped when given more time: WASP+DLV achieved a score of 2200, closely followed by ME-ASP with score 2190, and then LP2NORMAL+CLASP with score 1952.

Modeling with ASP is Fun!

Following the positive experience from 2014, a second on-site modeling event was held at LPNMR 2015. Five teams took the challenge of devising encodings for as many as they could among five modeling tasks, targeting discrete structures and logic puzzles, within a 2-hour session.

The first place went to a team from TU Vienna: M. Abseher and G. Charwat succeeded to solve three of the given tasks. A mixed three-member team took the second place: C. Dodaro (University of Calabria), R. Kaminski (University of Potsdam), and K. Shchekotykhin (Alpen-Adria University of Klagenfurt) also solved three tasks, yet with longer accumulated submission time. The third place was won by a mixed team as well, including one of the inventors of ASP: A. Harrison and V. Lifschitz (University of Texas at Austin) together with B. Susman (University of Nebraska at Omaha) completed two of the five modeling tasks.

Conclusion

ASP competitions started in 2007 and influenced the evolution of the ASP community by posing new challenges and showcasing achievements. So, what's hot in ASP competitions? The advancement of the state of the art in ASP solving was consistent and continuous in the last editions. A variety of application-oriented benchmarks, challenging system developers and reflecting the progress of implementations, are nowadays available. This is complemented by on-site modeling events to attract researchers and students to use ASP, emphasizing the fun of it while keeping the extra effort low.

Acknowledgments

This work was partially funded by MIUR under PON project "SI-LAB BA2KNOW – Business Analytics to Know", by Regione Calabria, programme POR Calabria FESR 2007–2013, and projects "ITravel PLUS" and "KnowRex: Un sistema per il riconoscimento e lestrazione di conoscenza".

References

Alviano, M.; Dodaro, C.; Leone, N.; and Ricca, F. 2015. Advances in WASP. In *Logic Programming and Nonmonotonic Reasoning – 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *LNCS*, 40–54. Springer.

Alviano, M.; Dodaro, C.; and Ricca, F. 2014. Anytime computation of cautious consequences in answer set programming. *Theory and Practice of Logic Programming* 14(4-5):755–770.

Bomanson, J., and Janhunen, T. 2013. Normalizing cardinality rules using merging and sorting constructions. In *Logic Programming and Nonmonotonic Reasoning – 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings*, volume 8148 of *LNCS*, 187–199. Springer.

Bomanson, J.; Gebser, M.; and Janhunen, T. 2014. Improving the normalization of weight rules in answer set programs. In *Logics in Artificial Intelligence – 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, volume 8761 of *LNCS*, 166–180. Springer.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12):92–103.

Calimeri, F.; Ianni, G.; Krennwallner, T.; and Ricca, F. 2012. The answer set programming competition. *AI Magazine* 33(4):114–118.

Calimeri, F.; Gebser, M.; Maratea, M.; and Ricca, F. 2015. Design and results of the fifth answer set programming competition. *Artificial Intelligence*. In press, available online: `http://www.sciencedirect.com/science/article/pii/S0004370215001447`.

Calimeri, F.; Ianni, G.; and Ricca, F. 2014. The third open answer set programming competition. *Theory and Practice of Logic Programming* 14(1):117–135.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Romero, J.; and Schaub, T. 2015. Progress in clasp series 3. In *Logic Programming and Nonmonotonic Reasoning – 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *LNCS*, 368–383. Springer.

Gebser, M.; Maratea, M.; and Ricca, F. 2015. The design of the sixth answer set programming competition. In *Logic Programming and Nonmonotonic Reasoning – 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *LNCS*, 531–544. Springer.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7(3):499–562.

Maratea, M.; Pulina, L.; and Ricca, F. 2014. A multi-engine approach to answer-set programming. *Theory and Practice of Logic Programming* 14(6):841–868.

Maratea, M.; Pulina, L.; and Ricca, F. 2015. Multi-level algorithm selection for ASP. In *Logic Programming and Nonmonotonic Reasoning – 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *LNCS*, 439–445. Springer.