DLV: Evolution and Perspectives

Mario Alviano[1], Francesco Calimeri[1](✉), Carmine Dodaro[2],
Davide Fuscà[1], Nicola Leone[1], Simona Perri[1],
Francesco Ricca[1], Pierfrancesco Veltri[3], and Jessica Zangari[1]

[1] DeMaCS, University of Calabria, Italy, {lastname}@mat.unical.it
[2] DIBRIS, University of Genova, Italy, dodaro@dibris.unige.it
[3] DLVSystem Srl, Italy, veltri@dlvsystem.com

Abstract. DLV is a system for Answer Set Programming (ASP), a logic-based programming paradigm for solving problems in a fully declarative way. It has been one of the first solid and reliable ASP systems, widely used in academy and fruitfully employed in many relevant industrial applications. In this paper we first provide an overview of DLV and its evolution, and then present DLV2 [1], a completely re-engineered version of the system that relies on the combination of the $\mathcal{I}$-DLV grounder [2] and the WASP solver [3]. The new system supports ASP-Core-2 input programs that can be enriched by annotations and directives for customizing heuristics of the system and extending solving capabilities; furthermore, an empirical analysis shows that DLV2 also outperforms the "old" DLV in terms of performance.

Keywords: Answer Set Programming, Systems, Grounding, Solving.

1 Introduction

Answer Set Programming (ASP) [4, 5] is a declarative programming paradigm proposed in the area of non-monotonic reasoning and logic programming. In ASP, computational problems are encoded by logic programs whose answer sets, corresponding to solutions, are computed by an ASP system [6]. Throughout the years, the availability of reliable high-performance implementations [7] made ASP a powerful tool for developing advanced applications in many research areas, ranging from Artificial Intelligence to Databases and Bioinformatics, as well as in industrial contexts [7–11].

ASP systems rely on a grounding module (*grounder*), that generates a propositional theory semantically equivalent to the input program, coupled with a subsequent module (*solver*) that applies propositional techniques for generating its answer sets [12]. The inherent complexity of engineering a *monolithic* (i.e., including both grounder and solver in the same implementation) ASP system favored the development of stand-alone grounders and solvers [13]; however, the requirements of recent ASP applications have been renewing the interest in integrated solutions. Indeed, monolithic systems can offer more control over the grounding and solving process, a more tight integration between the two modules that can also lead to performance gains, and more flexibility in developing application-oriented features.

In this paper we overview DLV [14], one of the first solid monholitic ASP systems. We first outline its features and applications, that have been developed through the years, and then focus on DLV2 [1], a completely renewed version of the system. In particular, we illustrate the new features of DLV2 and present an empirical analysis conducted on benchmarks from past ASP competitions, showing that it substantially improves over the old system in terms of performance. Eventually, we discuss ongoing and future work.

2 DLV: History and Applications

DLV [14] has been one of the first monolithic ASP systems. The DLV project started in the 90-ties (it started a few years after the first definition of answer set semantics [15]), and encompassed the development and the continuous enhancements of the system. This led to the integration of most of the theoretical advancements in ASP language and computation, thus offering advanced knowledge modeling features for effective and efficient declarative problem solving [16–24]. In addition, the system incorporated several front-ends for dealing with specific applications and featured tools for developing ASP-based solutions [25–28].

Over the years DLV proved to be suitable for both academic and real-world applications, and it has been widely employed in both scenarios, significantly contributing in spreading the use of ASP. As a matter of example, we mention here that it has been used for educational purposes in courses on Databases and Artificial Intelligence, both in European and American universities; it has been employed at CERN, the European Laboratory for Particle Physics, for a deductive database application; the Polish company Rodan Systems S.A. has released a DLV-based tool actually used for the detection of price manipulations and unauthorized use of confidential information by the Polish Securities and Exchange Commission; the European Commission funded a project on Information Integration, which produced a sophisticated and efficient data integration system based on it, called INFOMIX [29].

In addition, DLV has stimulated quite some interest in industry, also fostering technological transfer activities: several DLV-based applications have been released by two spin-off companies of the University of Calabria: EXEURA and DLVSystem. In the following, we report some interesting examples of industry-level applications of DLV; further details and other applications can be found in the literature (see, e.g., [8, 30]).

Workforce Management. DLV has been employed for a workforce management [10] at the Gioia Tauro seaport – the largest transhipment port of the Mediterranean Sea, in order to compute a suitable allocation of the available personnel such that cargo ships mooring in the port are properly handled, by complying to a significant number of non-trivial constraints.

E-Tourism. Several DLV-based applications have been developed in the tourism industry [31]; for instance, an intelligent advisor has been created, that as a mediator finds the best matches between offers and customers, thus reducing the time to single out and sell touristic offers and increasing customer satisfaction.
Intelligent Call Routing. A platform for intelligent phone call routing has been developed in a production system handling Telecom Italia call-centers; it classifies customer profiles and allows to anticipate their actual needs, thus creating a personalized experience of customer care service. The operators assignment process is significantly optimized, both reducing response times and improving customer support quality.
E-Medicine. Medical knowledge bases, resulting from the integration of different databases, often suffer from errors and anomalies; a multi-source data cleaning system, based on DLV, has been implemented, that automatically generates ad-hoc ASP programs for identifying and possibly fixing both syntactic and semantic anomalies in such knowledge bases.

3 Evolution: DLV2

After years of incremental updates, DLV has been completely renewed; in particular, still relying on several of its solid theoretical foundations, it has been redesigned and re-engineered, aiming at building a modern ASP system featuring new evaluation techniques and endowed with proper development tools. In particular, the project first focussed on developing separate solutions for *grounding* and *solving* [12], releasing the $\mathcal{I}$-DLV grounder [2] and the WASP solver [3]; later on, these have been integrated in a monolithic system called DLV2 [1]. Besides enjoying the features of both subsystems, DLV2 takes advantage from a tight and efficient coupling of them in terms of performance, usability and ease of integration into ASP-based applications. DLV2 is fully compliant with ASP-Core-2, the standard language that the ASP community released years after the first DLV release in order to harmonize the dialects supported by different ASP implementations [32]; furthermore, it offers additional constructs for further enhancing usability in real-world contexts, such as lists.

Since its first version, DLV constituted a full-fledged deductive-database system; besides explicit means for accessing external databases, it incorporates a number of ad-hoc techniques for handling advanced database applications, such as join-decomposition methods, magic-sets and other DB-oriented features [2]. DLV2 maintains this capabilities, and significantly extends them, allowing the access to data not only from relational databases but also from graph databases for easing applications in semantic web contexts.

Furthermore, DLV2 explicitly features proper mechanisms for easing interoperability and integration with external systems, that are crucial for the development of real-world and industrial applications of ASP. For instance, besides the abovementioned native directives for importing/exporting data from relational and graph databases, a framework is available for integrating calls to custom Python scripts via external atom, enabling the definition of custom interoper-

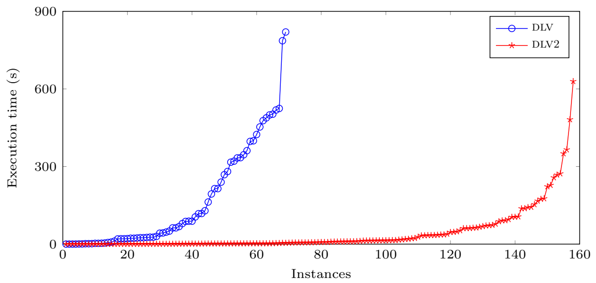

Fig. 1: DLV2 vs DLV: cactus plot on benchmarks from ASP Competition 2011.

ability means, further extensions of system capabilities and the accommodation of external sources of computation and value invention within ASP programs [2].

The DLV2 system is very flexible, allowing to tailor its behaviour to the problem at hand and properly handle specific properties of ASP programs. We mention here the support for external propagators, the definition of new heuristics [33], and novel constructs, such as annotations within the ASP code, for granting a fine-grained control over the whole computational process [1].

Besides the development of ASP systems, one of the most relevant aspects of the DLV project over the years has been the release of tools for development and software engineering, in order to foster practically viable ASP-based applications. With this respect, tools that were already released for the use with DLV are being updated for DLV2, and additional new solutions are being developed; we mention here: ASPIDE [34], a comprehensive framework for advanced program composition and execution; JDLV, a framework for integrating ASP programs within Java [28]; LoIDE [35], a web-based IDE for ASP; EmbASP, a general framework for easing the development of applications embedding ASP [36].

As it emerges from the discussion above, DLV2 constitutes a significant advancement with respect to its predecessor, in terms of features and usability. Nonetheless, significant improvements can be observed also when performance is considered. In order to show this, we report the results of an experimental comparison between the old and the new system, both tested with their default configuration. Since DLV does not fully support the ASP-Core-2 standard language, the comparison has been performed on benchmarks taken from the third ASP competition [37], the latest featuring domains encoded with ASP programs supported by DLV. Experiments were performed on a NUMA machine equipped with two 2.8`GHz AMD Opteron 6320` processors. The time and memory were limited to 900 seconds and 15 GB, respectively. Results are reported in the cactus plot in Figure 1. Without going too much in details, the performance boost enjoyed by DLV2 is evident: the percentage gain of solved instances is 128%; it is even higher (273%) if running time is bounded to 60 seconds.

4 Perspectives: Beyond DLV2

In this paper we presented DLV2, the new version of the ASP system DLV that properly combines the $\mathcal{I}$-DLV grounder [2] and the WASP solver [3] and results as significantly enhanced with respect to its predecessor both in terms of features and performance. The DLV project is currently focussing on further improving DLV2 performance, and also enriching it with means for properly addressing the challenges arising from the application of ASP in the emerging contexts of Big Data and Stream Reasoning.

The system is available online at `https://www.mat.unical.it/DLV2`.

Acknowledgments. The work has been partially supported by the Italian ministry for economic development (MISE) under project "PIUCultura - Paradigmi Innovativi per l'Utilizzo della Cultura" (n. F/020016/01-02/X27), and under project "Smarter Solutions in the Big Data World (S2BDW)".

References

1. Alviano, M., Calimeri, F., Dodaro, C., Fuscà, D., Leone, N., Perri, S., Ricca, F., Veltri, P., Zangari, J.: The ASP system DLV2. In: LPNMR. Lecture Notes in Computer Science, vol. 10377, pp. 215–221. Springer (2017)
2. Calimeri, F., Fuscà, D., Perri, S., Zangari, J.: I-DLV: the new intelligent grounder of DLV. Intelligenza Artificiale 11(1), 5–20 (2017), `http://dx.doi.org/10.3233/IA-170104`
3. Alviano, M., Dodaro, C., Leone, N., Ricca, F.: Advances in WASP. In: LPNMR. LNCS, vol. 9345, pp. 40–54. Springer (2015)
4. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Commun. ACM 54(12), 92–103 (2011)
5. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. 9(3/4), 365–386 (1991)
6. Lifschitz, V.: Answer set planning. In: ICLP. pp. 23–37. MIT Press (1999)
7. Calimeri, F., Gebser, M., Maratea, M., Ricca, F.: Design and results of the fifth answer set programming competition. Artif. Intell. 231, 151–181 (2016)
8. Leone, N., Ricca, F.: Answer set programming: A tour from the basics to advanced development tools and industrial applications. In: Faber, W., Paschke, A. (eds.) Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures. Lecture Notes in Computer Science, vol. 9203, pp. 308–326. Springer (2015), `https://doi.org/10.1007/978-3-319-21768-0_10`
9. Nogueira, M., Balduccini, M., Gelfond, M., Watson, R., Barry, M.: An a-prolog decision support system for the space shuttle. In: Ramakrishnan, I.V. (ed.) Practical Aspects of Declarative Languages, Third International Symposium, PADL 2001, Las Vegas, Nevada, March 11-12, 2001, Proceedings. Lecture Notes in Computer Science, vol. 1990, pp. 169–183. Springer (2001), `https://doi.org/10.1007/3-540-45241-9_12`
10. Ricca, F., Grasso, G., Alviano, M., Manna, M., Lio, V., Iiritano, S., Leone, N.: Team-building with answer set programming in the gioia-tauro seaport. TPLP 12(3), 361–381 (2012)

11. Tiihonen, J., Soininen, T., Niemelä, I., Sulonen, R.: A practical tool for mass-customising configurable products. In: DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm (2003)
12. Kaufmann, B., Leone, N., Perri, S., Schaub, T.: Grounding and solving in answer set programming. AI Magazine 37(3), 25–32 (2016), http://www.aaai.org/ojs/index.php/aimagazine/article/view/2672
13. Lierler, Y., Maratea, M., Ricca, F.: Systems, engineering environments, and competitions. AI Magazine 37(3), 45–52 (2016), http://www.aaai.org/ojs/index.php/aimagazine/article/view/2675
14. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. 7(3), 499–562 (2006)
15. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing 9(3/4), 365–385 (1991), http://dx.doi.org/10.1007/BF03037169
16. Calimeri, F., Cozza, S., Ianni, G., Leone, N.: Computable Functions in ASP: Theory and Implementation. In: Proceedings of the 24th International Conference on Logic Programming (ICLP 2008). vol. 5366, pp. 407–424. Springer, Udine, Italy (Dec 2008)
17. Calimeri, F., Ianni, G.: Template programs for disjunctive logic programming: An operational semantics. AI Commun. 19(3), 193–206 (2006), http://content.iospress.com/articles/ai-communications/aic373
18. Calimeri, F., Citrigno, M., Cumbo, C., Faber, W., Leone, N., Perri, S., Pfeifer, G.: New DLV features for data integration. In: Alferes, J.J., Leite, J.A. (eds.) Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, Lisbon, Portugal, September 27-30, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3229, pp. 698–701. Springer (2004), https://doi.org/10.1007/978-3-540-30227-8_60
19. Alviano, M., Calimeri, F., Faber, W., Leone, N., Perri, S.: Unfounded sets and well-founded semantics of answer set programs with aggregates. CoRR abs/1401.4596 (2014), http://arxiv.org/abs/1401.4596
20. Perri, S., Scarcello, F., Catalano, G., Leone, N.: Enhancing DLV instantiator by backjumping techniques. Ann. Math. Artif. Intell. 51(2-4), 195–228 (2007), https://doi.org/10.1007/s10472-008-9090-9
21. Calimeri, F., Perri, S., Ricca, F.: Experimenting with parallelism for the instantiation of ASP programs. J. Algorithms 63(1-3), 34–54 (2008), https://doi.org/10.1016/j.jalgor.2008.02.003
22. Perri, S., Leone, N.: Parametric connectives in disjunctive logic programming. AI Commun. 17(2), 63–74 (2004), http://content.iospress.com/articles/ai-communications/aic308
23. Buccafurri, F., Leone, N., Rullo, P.: Enhancing disjunctive datalog by constraints. IEEE Trans. Knowl. Data Eng. 12(5), 845–860 (2000), https://doi.org/10.1109/69.877512
24. Alviano, M., Faber, W., Greco, G., Leone, N.: Magic sets for disjunctive datalog programs. Artif. Intell. 187, 156–192 (2012)
25. Citrigno, S., Eiter, T., Faber, W., Gottlob, G., Koch, C., Leone, N., Mateis, C., Pfeifer, G., Scarcello, F.: The dlv system: Model generator and advanced frontends (system description). In: WLP. p. 0 (1997)
26. Fuscà, D., Germano, S., Zangari, J., Anastasio, M., Calimeri, F., Perri, S.: A framework for easing the development of applications embedding answer set programming. In: Cheney, J., Vidal, G. (eds.) Proceedings of the 18th Interna-

tional Symposium on Principles and Practice of Declarative Programming, Edinburgh, United Kingdom, September 5-7, 2016. pp. 38–49. ACM (2016), http://doi.acm.org/10.1145/2967973.2968594

27. Febbraro, O., Leone, N., Reale, K., Ricca, F.: Extending ASPIDE with user-defined plugins. In: Lisi, F.A. (ed.) Proceedings of the 9th Italian Convention on Computational Logic, Rome, Italy, June 6-7, 2012. CEUR Workshop Proceedings, vol. 857, pp. 236–240. CEUR-WS.org (2012), http://ceur-ws.org/Vol-857/paper_s01.pdf
28. Febbraro, O., Leone, N., Grasso, G., Ricca, F.: JASP: A framework for integrating answer set programming with java. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012. AAAI Press (2012), http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4520
29. Leone, N., Greco, G., Ianni, G., Lio, V., Terracina, G., Eiter, T., Faber, W., Fink, M., Gottlob, G., Rosati, R., Lembo, D., Lenzerini, M., Ruzzi, M., Kalka, E., Nowicki, B., Staniszkis, W.: The INFOMIX system for advanced integration of incomplete and inconsistent data. In: Özcan, F. (ed.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005. pp. 915–917. ACM (2005), http://doi.acm.org/10.1145/1066157.1066286
30. Erdem, E., Gelfond, M., Leone, N.: Applications of answer set programming. AI Magazine 37(3), 53–68 (2016)
31. Ricca, F., Dimasi, A., Grasso, G., Ielpa, S.M., Iiritano, S., Manna, M., Leone, N.: A logic-based system for e-tourism. Fundam. Inform. 105(1-2), 35–55 (2010)
32. Calimeri, F., Faber, W., Gebser, M., Ianni, G., Kaminski, R., Krennwallner, T., Leone, N., Ricca, F., Schaub, T.: Asp-core-2: Input language format. ASP Standardization Working Group, Techical Report (2012)
33. Dodaro, C., Gasteiger, P., Leone, N., Musitsch, B., Ricca, F., Schekotihin, K.: Combining answer set programming and domain heuristics for solving hard industrial problems (application paper). TPLP 16(5-6), 653–669 (2016)
34. Febbraro, O., Reale, K., Ricca, F.: ASPIDE: integrated development environment for answer set programming. In: LPNMR. Lecture Notes in Computer Science, vol. 6645, pp. 317–330. Springer (2011)
35. Germano, S., Calimeri, F., Palermiti, E.: Loide: A web-based IDE for logic programming preliminary report. In: Calimeri, F., Hamlen, K.W., Leone, N. (eds.) Practical Aspects of Declarative Languages - 20th International Symposium, PADL 2018, Los Angeles, CA, USA, January 8-9, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10702, pp. 152–160. Springer (2018), https://doi.org/10.1007/978-3-319-73305-0_10
36. Fuscà, D., Germano, S., Zangari, J., Anastasio, M., Calimeri, F., Perri, S.: A framework for easing the development of applications embedding answer set programming. In: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, Edinburgh, United Kingdom, September 5-7, 2016. pp. 38–49 (2016), http://doi.acm.org/10.1145/2967973.2968594
37. Calimeri, F., Ianni, G., Ricca, F.: The third open answer set programming competition. TPLP 14(1), 117–135 (2014), http://dx.doi.org/10.1017/S1471068412000105