LNAI 13416

Georg Gottlob
Daniela Inclezan
Marco Maratea (Eds.)

Logic Programming and Nonmonotonic Reasoning

16th International Conference, LPNMR 2022
Genova, Italy, September 5–9, 2022
Proceedings

LPNMR 22
16th Int. Conference
Logic Programming
Nonmonotonic Reasoning

Springer

Lecture Notes in Artificial Intelligence 13416

Subseries of Lecture Notes in Computer Science

More information about this subseries at https://link.springer.com/bookseries/1244

Georg Gottlob · Daniela Inclezan ·
Marco Maratea (Eds.)

Logic Programming and Nonmonotonic Reasoning

16th International Conference, LPNMR 2022
Genova, Italy, September 5–9, 2022
Proceedings

Springer

Editors
Georg Gottlob
University of Oxford
Oxford, UK

Daniela Inclezan
Miami University
Oxford, OH, USA

Marco Maratea
University of Genoa
Genoa, Italy

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-15706-6 ISBN 978-3-031-15707-3 (eBook)
https://doi.org/10.1007/978-3-031-15707-3

LNCS Sublibrary: SL7 – Artificial Intelligence

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the papers presented at the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022) held during September 5–9, 2022, in Genova Nervi, Italy.

LPNMR 2022 was the sixteenth in the series of international meetings on logic programming and non-monotonic reasoning. LPNMR is a forum for exchanging ideas on declarative logic programming, non-monotonic reasoning, and knowledge representation. The aim of the conference is to facilitate interactions between researchers and practitioners interested in the design and implementation of logic-based programming languages and database systems, and those working in knowledge representation and non-monotonic reasoning. LPNMR strives to encompass theoretical and experimental studies that have led or will lead to advances in declarative programming and knowledge representation, as well as their use in practical applications. The past editions of LPNMR were held in Washington, D.C., USA (1991), Lisbon, Portugal (1993), Lexington, Kentucky, USA (1995), Dagstuhl, Germany (1997), El Paso, Texas, USA (1999), Vienna, Austria (2001), Fort Lauderdale, Florida, USA (2004), Diamante, Italy (2005), Tempe, Arizona, USA (2007), Potsdam, Germany (2009), Vancouver, Canada (2011), Coruña, Spain (2013), Lexington, Kentucky, USA (2015), Espoo, Finland (2017), and Philadelphia, USA (2019).

LPNMR 2022 received 57 submissions. Every submission was reviewed by at least three Program Committee members. In total, 34 papers were accepted as regular long papers, and five as short papers. Thus, 39 of the 57 papers were accepted. The scientific program also included four invited talks by Nicola Leone, University of Calabria, Italy; Sheila McIlraith, University of Toronto, Canada; Alessandra Russo, Imperial College London, UK; and Stefan Woltran, TU Wien, Austria. Moreover, the program was completed by three thematic invited tutorials by Stefania Costantini, University of L'Aquila, Italy; Viviana Mascardi, University of Genoa, Italy; and Andreas Pieris, University of Edinburgh, UK.

Springer sponsored the best technical paper award, while the Italian Association for Logic Programming (GULP) sponsored for the best student paper award. These awards were granted during the conference, followed by the selection of papers to have their long versions invited for Rapid Publication Track to the Artificial Intelligence Journal and to the journal of Theory and Practice of Logic Programming.

Three workshops were co-located with LPNMR 2022: the 4th International Workshop on the Resurgence of Datalog in Academia and Industry (DATALOG 2.0), the First International Workshop on HYbrid Models for Coupling Deductive and Inductive ReAsoning (HYDRA 2022), and the 29th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2022). A Doctoral Consortium (DC) was also part of the program. We thank the workshop and DC organizers for their efforts.

We would like to express our warmest thanks and acknowledgments to those who played an important role in the organization of LPNMR 2022: the Program Committee and additional reviewers for their fair and thorough evaluations of submitted papers; Viviana Mascardi for coordinating the workshops, Martin Gebser for organizing the Doctoral Consortium, Jessica Zengari for advertising the conference and its workshops through a number of channels, and the members of the Local Organizing Committee (Angelo Ferrando, Matteo Cardellini, and Marco Mochi) and the other volunteer members for working hard towards the success of the event.

The LPNMR 2022 conference received support from several organizations. We gratefully acknowledge the DIBRIS Department of the University of Genoa, the National Science Foundation, the Artificial Intelligence Journal, the Italian Association for Logic Programming (GULP), Springer, the Association for Logic Programming, Potassco Solutions, SurgiQ, DLVSystem, the Royal Society (supporting G. Gottlob by Project RAISON DATA No. RP\R1\201074), and the Alan Turing Institute.

The conference was managed with the help of EasyChair.

September 2022

Georg Gottlob
Daniela Inclezan
Marco Maratea

Organization

General Chair

Georg Gottlob	University of Oxford, UK

Program Committee Chairs

Daniela Inclezan	Miami University, USA
Marco Maratea	University of Genoa, Italy

Workshop Chair

Viviana Mascardi	University of Genoa, Italy

Doctoral Consortium Chair

Martin Gebser	University of Klagenfurt, Austria

Publicity Chair

Jessica Zangari	University of Calabria, Italy

Local Organization

Angelo Ferrando	University of Genoa, Italy
Matteo Cardellini	University of Genoa, Italy
Marco Mochi	University of Genoa, Italy

Program Committee

Marco Alviano	University of Calabria, Italy
Chitta Baral	Arizona State University, USA
Marcello Balduccini	Saint Joseph's University, USA
Francesco Calimeri	University of Calabria, Italy
Marina De Vos	University of Bath, UK
Carmine Dodaro	University of Calabria, Italy
Agostino Dovier	University of Udine, Italy
Thomas Eiter	TU Wien, Austria
Esra Erdem	Sabanci University, Turkey
Wolfgang Feber	University of Klagenfurt, Austria
Johannes K. Fichte	TU Wien, Austria
Paul Fodor	Stony Brook University, USA

Gerhard Friedrich	University of Klagenfurt, Austria
Sarah Alice Gaggl	TU Dresden, Germany
Martin Gebser	University of Klagenfurt, Austria
Michael Gelfond	Texas Tech University, USA
Gopal Gupta	University of Texas at Dallas, USA
Katsumi Inoue	National Institute of Informatics, Japan
Tomi Janhunen	Tampere University, Finland
Gabriele Kern-Isberner	TU Dortmund, Germany
Matthias Knorr	Universidade Nova de Lisboa, Portugal
Emily LeBlanc	Drexel University, USA
João Leite	Universidade Nova de Lisboa, Portugal
Vladimir Lifschitz	University of Texas at Austin, USA
Enrico Pontelli	New Mexico State University, USA
Francesco Ricca	University of Calabria, Italy
Alessandra Russo	Imperial College London, UK
Orkunt Sabuncu	TED University, Turkey
Chiaki Sakama	Wakayama University, Japan
Torsten Schaub	University of Potsdam, Germany
Guillermo R. Simari	Universidad del Sur in Bahia Blanca, Argentina
Theresa Swift	Universidade Nova de Lisboa, Portugal
Daniele Theseider Dupré	Università del Piemonte Orientale, Italy
Matthias Thimm	FernUniversität in Hagen, Germany
Hans Tompits	TU Wien, Austria
Mirek Truszczynski	University of Kentucky, USA
Johannes P. Wallner	Graz University of Technology, Austria
Kewen Wang	Griffith University, Australia
Stefan Woltran	TU Wien, Austria
Jia-Huai You	University of Alberta, Canada

Additional Reviewers

Ramsha Ali
George Baryannis
Sotiris Batsakis
Arvid Becker
Michael Bernreiter
Elise Böhl
Martin Diller
Dragan Doder
Mohammed El-Kholany
Tobias Geibinger
Susana Hahn
Markus Hecher
Antonio Ielo
Rafael Kiesel
Giovanni Melissari
Johannes Oetsch
Henry Otunuya
Kristian Reale
Dominik Rusovac
Elmer Salazar
Emanuel Sallinger
Patrik Schneider
Kenneth Skiba
Klaus Strauch
Alice Tarzariol
Sarat Chandra Varanasi
Zhiqiang Zhuang

Sponsors and Collaborators

Alan Turing Institute
Artificial Intelligence Journal, Elsevier
Association for Logic Programming (ALP)
DIBRIS, University of Genoa
DLVSystem
Gruppo Utenti Logic Programming (GULP)
Italian Association for Artificial Intelligence (AI*IA)
National Science Foundation (NSF)
Potassco Solutions
Royal Society
Springer
SurgiQ
Theory and Practice of Logic Programming, Cambridge University Press

Abstracts of Invited Talks

DLV Evolution from Datalog to Ontology and Stream Reasoning

N. Leone, M. Alviano, F. Calimeri, C. Dodaro, G. Ianni, M. Manna, E. Mastria, M.C. Morelli, F. Pacenza, S. Perri, K. Reale, F. Ricca, G. Terracina, and J. Zangari

University of Calabria, Italy
{n.leone, m.alviano, f.calimeri, c.dodaro, g.ianni, m.manna, e.mastria, m.c.morelli, f.pacenza, s.perri, k.reale, f.ricca, g.terracina, j.zangari}@unical.it

Abstract. DLV has been one of the first solid and reliable integrated systems for Answer Set Programming (ASP). DLV has significantly contributed both in spreading the use of ASP and in fostering AI-based technological transfer activities. This paper overviews the history and the recent evolution of the system, which enable effective reasoning on ontologies and streams of data, and the development of new applications.

Keywords: Answer set programming · Ontologies · Stream reasoning

The DLV System

DLV [27] has been one of the first solid and reliable integrated ASP systems. Its project started a few years after the first definition of answer set semantics [21]. It has always been, since the beginning, a suitable tool for applications in academic and real-world scenarios, and significantly contributed both in spreading the use of ASP and in fostering AI-based technological transfer activities [2, 19, 23]. After years of incremental updates, a brand new version has been released, namely DLV-2 [4], a modern ASP system featuring efficient evaluation techniques, proper development tools, versatility, and interoperability. The project firstly focussed on developing separate solutions for *grounding* and *solving*, releasing the I-DLV grounder [13] and the WASP solver [5]; later on, the two systems have been integrated in a monolithic, yet slender body. As for the input language, DLV-2 was born fully compliant with the ASP-Core-2 standard language; in addition, it offers additional constructs and tools for further enhancing usability in real-world contexts [2, 23]. Historically, one of the most distinctive traits of DLV is a full-fledged deductive-database system nature; nevertheless, it has been steadily maintained and properly updated beyond this scope to handle an increasing number of real-world and industrial applications. Actually, the development of industrial applications of DLV started around 2010, with the first success story being the development of a team-building system [23]. The number of DLV-based industrial applications is constantly growing, among latest we mention: a system querying

DBpedia in natural language [17], a tool for rescheduling of nurse shifts in hospitals [6], a decision support system for the diagnosis of headache disorders [16], and a system for compliance-checking of electric panels [8]. Recently, DLV has been empowered with tools and extensions to handle large scale reasoning with Datalog, run on smart devices, and connect to big data systems [26, 28]. Nonetheless, some of the most compelling challenges consist of empowering DLV with means for ontological reasoning, and stream reasoning.

DLV for Ontological Reasoning

Since 2012, DLV has been actively supporting O*ntology-Based Query Answering* (OBQA) [10], where a query $q(\mathbf{x})$ is evaluated over a *knowledge base* consisting of an extensional *dataset D* paired with an *ontology* Σ. In this context, Description Logics (DLs) [1] and Datalog$^{\pm}$ [10] have been recognized as the two main formalisms to specify ontologies. Unfortunately, in both cases, OBQA is generally undecidable [9]. To overcome this limitation, a number of classes of ontologies that guarantee the decidability of query answering have been proposed with the aim of offering a good balance between computational complexity and expressiveness. Since DLV natively deals with plain Datalog, it can deal with Linear [11], Guarded [9] and Sticky [12], which are Datalog rewritable under conjunctive queries, namely the ontology and the query can be rewritten, independently from datasets, into an equivalent Datalog program. Analogously, since DLV natively supports function symbols and value invention in a controlled way, it directly supports Weakly−Acyclic, which admits canonical models of finite size. In 2012, DLV started supporting Shy, which encompasses and generalizes plain Datalog, Linear and DL-Lite$_{\mathsf{R}}$. In particular, DLV$^{\exists}$ [25]—the branch of DLV supporting Shy—implements a fixed-point operator called *parsimonious chase* and it is still considered a top system over these classes [7]. Subsequently, a new branch of DLV, called OWL2DLV [3], has been developed with the aim of evaluating SPARQL queries over very large OWL 2 knowledge bases. In particular, OWL2DLV supports Horn-$\mathcal{SHIQ}$ and a large fragment of $\mathcal{EL}$++. Moreover, OWL2DLV incorporates novel optimizations sensibly reducing memory consumption and a server-like behavior to support multiple query scenarios. The high potential of OWL2DLV for large-scale reasoning is outlined by the results of an experiment on data-intensive benchmarks, and confirmed by the direct interest of a major international industrial player, which has stimulated and partially supported this work. More recently, DaRLing [20]—a Datalog rewriter for DLP ontologies under SPARQL queries—enriched the DLV suite for OBQA to deal with the *sameAs* and to support concrete datatypes. Finally, by exploiting a novel algorithm designed for the so called *dyadic existential rules* [22], it is now possible to exploit DLV$^{\exists}$ to deal also with Ward ontologies.

DLV for Stream Reasoning

DLV has been empowered with Stream Reasoning capabilities, which are nowadays a key requirement for deploying effective applications in several real-world domains, such as IoT, Smart Cities, Emergency Management. Stream Reasoning (SR) [18] consists in the application of inference techniques to highly dynamic data streams, and ASP is generally acknowledged as a particularly attractive basis for it. In this view, a new incarnation of DLV has been released, namely *I-DLV-sr* [15], that features a language ad-hoc conceived for easily modeling SR tasks along with robust performance and high scalability. In fact, the input language consists in normal (i.e., non-disjunctive) stratified ASP programs featuring *streaming* literals in rule bodies, over the operators: *in*, *always*, *count*, *at least*, and *at most*; recursion involving streaming literals is allowed. The system takes advantage from *Apache Flink* for efficiently processing data streams and from incremental evaluation techniques [14, 24] to efficiently scale over real-world application domains. *I-DLV-sr* proved to be effectively usable over real-world SR domains; nevertheless, being under steady development, it has been significantly improving over time with respect of stability, performance and language features; for instance, inspired by applications in the smart city domain, new constructs have been recently introduced that further ease the modeling of reasoning tasks and enable new functionalities, such as external sources of computation, trigger rules, means for explicitly refer to time, generalized streaming atoms.

Conclusion

DLV is one of the first solid and reliable integrated ASP systems. We reported on the development of DLV, mentioned some of the latest applications, and focused on some recent enhancements for reasoning on ontologies and streams of data.

References

1. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Adrian, W.T., et al.: The ASP system DLV: advancements and applications. KI. **32,** 177–179 (2018). https://doi.org/10.1007/s13218-018-0533-0
3. Allocca, C., et al.: Large-scale reasoning on expressive horn ontologies. In: Datalog. CEUR Workshop Proceedings, vol. 2368, pp. 10–21. CEUR-WS.org (2019)
4. Alviano, M., et al.: The ASP system DLV2. In: Balduccini, M., Janhunen, T. (eds.) LPNMR 2017. LNCS, vol. 10377, pp. 215–221. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61660-5_19
5. Alviano, M., Dodaro, C., Leone, N., Ricca, F.: Advances in WASP. In: Calimeri, F., Ianni, G., Truszczynski, M. (eds.) LPNMR 2015. LNCS, vol. 9345, pp. 40–54. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23264-5_5
6. Alviano, M., Dodaro, C., Maratea, M.: Nurse (re)scheduling via answer set programming. Intelligenza Artificiale **12**(2), 109–124 (2018)
7. Baldazzi, T., Bellomarini, L., Favorito, M., Sallinger, E.: On the relationship between shy and warded datalog+/− (2022)
8. Barbara, V., et al.: Neural-symbolic approach to compliance of electric panels. In: Proceedings of the of CILC 22
9. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: query answering under expressive relational constraints. J. Artif. Intell. Res. **48**, 115–174 (2013)

10. Calì, A., Gottlob, G., Lukasiewicz, T.: Tractable query answering over ontologies with datalog+/−. In: Description Logics. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
11. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. J. Web Semant. **14**, 57–83 (2012)
12. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: the query answering problem. Artif. Intell. **193**, 87–128 (2012)
13. Calimeri, F., Fuscà, D., Perri, S., Zangari, J.: I-DLV: the new intelligent grounder of DLV. IA **11**(1), 5–20 (2017)
14. Calimeri, F., Ianni, G., Pacenza, F., Perri, S., Zangari, J.: Incremental answer set programming with overgrounding. TPLP **19**(5-6), 957–973 (2019)
15. Calimeri, F., Manna, M., Mastria, E., Morelli, M.C., Perri, S., Zangari, J.: I-dlv-sr: A stream reasoning system based on I-DLV. TPLP **21**(5), 610–628 (2021)
16. Costabile, R., Catalano, G., Cuteri, B., Morelli, M.C., Leone, N., Manna, M.: A logic-based decision support system for the diagnosis of headache disorders according to the ICHD-3 international classification. TPLP **20**(6), 864–879 (2020)
17. Cuteri, B., Reale, K., Ricca, F.: A logic-based question answering system for cultural heritage. In: Calimeri, F., Leone, N., Manna, M. (eds.) JELIA 2019. LNCS, vol. 11468, pp. 526–541. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19570-0_35
18. Dell'Aglio, D., Valle, E.D., van Harmelen, F., Bernstein, A.: Stream reasoning: a survey and outlook. Data Sci. **1**(1–2), 59–83 (2017)
19. Erdem, E., Gelfond, M., Leone, N.: Applications of answer set programming. AI Mag. **37**(3), 53–68 (2016)
20. Fiorentino, A., Zangari, J., Manna, M.: Darling: a datalog rewriter for OWL 2 RL ontological reasoning under SPARQL queries. TPLP **20**(6), 958–973 (2020)
21. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. NGC **9**(3/4), 365–386 (1991)
22. Gottlob, G., Manna, M., Marte, C.: Dyadic existential rules. In: Datalog 2.0. (2022, paper under review)
23. Grasso, G., Leone, N., Ricca, F.: Answer set programming: language, applications and development tools. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 19–34. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39666-3_3
24. Ianni, G., Pacenza, F., Zangari, J.: Incremental maintenance of overgrounded logic programs with tailored simplifications. TPLP **20**(5), 719–734 (2020)
25. Leone, N., Manna, M., Terracina, G., Veltri, P.: Fast query answering over existential rules. ACM Trans. Comput. Log. **20**(2), 12:1–12:48 (2019)
26. Leone, N., Perri, S., Ricca, F., Veltri, P., Zangari, J.: First steps towards reasoning on big data with DLV. In: Bergamaschi, S., Noia, T.D., Maurino, A. (eds.) Proceedings of the SEBD 2018. CEUR Workshop Proceedings, vol. 2161. CEUR-WS.org (2018)
27. Leone, N., et al.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. **7**(3), 499–562 (2006)
28. Reale, K., Calimeri, F., Leone, N., Ricca, F.: Smart devices and large scale reasoning via ASP: tools and applications. In: Cheney, J., Perri, S. (eds.) PADL 2022. LNCS, vol. 13165, pp. 154–161. Springer, Cham. https://doi.org/10.1007/978-3-030-94479-7_1

Reward Machines: Formal Languages and Automata for Reinforcement Learning

Sheila McIlraith

University of Toronto, Toronto, Canada
sheila@cs.toronto.edu

Reinforcement Learning (RL) is proving to be a powerful technique for building sequential decision making systems in cases where the complexity of the underlying environment is difficult to model. Two challenges that face RL are reward specification and sample complexity. Specification of a reward function – a mapping from state to numeric value – can be challenging, particularly when reward-worthy behaviour is complex and temporally extended. Further, when reward is sparse, it can require millions of exploratory episodes for an RL agent to converge to a reasonable quality policy. In this talk I'll show how formal languages and automata can be used to represent complex non-Markovian reward functions. I'll present the notion of a Reward Machine, an automata-based structure that provides a normal form representation for reward functions, exposing function structure in a manner that greatly expedites learning. Finally, I'll also show how these machines can be generated via symbolic planning or learned from data, solving (deep) RL problems that otherwise could not be solved.

Logic-Based Machine Learning: Recent Advances and Their Role in Neuro-Symbolic AI

Alessandra Russo, Mark Law, Daniel Cunnington, Daniel Furelos-Blanco, and Krysia Broda

Department of Computing, Imperial College London
{a.russo, mark.law09, d.cunnington20, d.furelos-blanco18, k.broda}@imperial.ac.uk

Abstract. Over the last two decades there has been a growing interest in logic-based machine learning, where the goal is to learn a logic program, called a *hypothesis*, that together with a given background knowledge explains a set of examples. Although logic-based machine learning has traditionally addressed the task of learning definite logic programs (with no negation), our logic-based machine learning approaches have extended this field to a wider class of formalisms for knowledge representation, captured by the answer set programming (ASP) semantics. The ASP formalism is truly declarative and due to its non-monotonicity it is particularly well suited to commonsense reasoning. It allows constructs such as choice rules, hard and weak constraints, and support for default inference and default assumptions. Choice rules and weak constraints are particularly useful for modelling human preferences, as the choice rules can represent the choices available to the user, and the weak constraints can specify which choices a human prefers. In the recent years we have made fundamental contributions to the field of logic-based machine learning by extending it to the learning of the full class of ASP programs and the first part of this talk provides an introduction to these results and to the general field of learning under the answer set semantics, referred here as *learning from answer sets* (LAS).

To be applicable to real-world problems, LAS has to be tolerant to noise in the data, scalable over large search spaces, amenable to user-defined domain-specific optimisation criteria and capable of learning interpretable knowledge from structured and unstructured data. The second part of this talk shows how these problems are addressed by our recently proposed FastLAS approach for learning Answer Set Programs, which is targeted at solving restricted versions of observational and non-observational predicate learning from answer sets tasks. The advanced features of our family of LAS systems have made it possible to solve a variety of real-world problems in a manner that is data efficient, scalable and robust to noise. LAS can be combined with statistical learning methods to realise neuro-symbolic solutions that perform both fast, low-level prediction from unstructured data, and high-level logic-based learning of interpretable knowledge. The talk concludes with presenting two such neuro-symbolic solutions for respectively solving image classification problems in the presence of distribution shifts, and discovering sub-goal structures for reinforcement learning agents.

Non-monotonic Logic-Based Machine Learning

Over the last decade we have witnessed a growing interest in Machine Learning. In recent years Deep Learning has been demonstrated to achieve high-levels of accuracy in data analytics, signal and information processing tasks, bringing transformative impact in domains such as facial, image, speech recognition, and natural language processing. They have best performance on computational tasks that involve large quantities of data and for which the labelling process and feature extraction would be difficult to handle. However, they suffer from two main drawbacks, which are crucial in the context of cognitive computing. They are not capable of supporting AI solutions that are good at more than one task. They are very effective when applied to single specific tasks, but applying the same technology from one task to another within the same class of problems would often require retraining, causing the system to possibly *forget* how to solve a previously learned task. Secondly, and most importantly, they are *not transparent*. Operating primarily as black boxes, deep learning approaches are not amenable to human inspection and human feedbacks, and the learned models are not explainable, leaving the humans agnostic of the cognitive and learning process performed by the system. This lack of transparency hinders human comprehension, auditing of the learned outcomes, and human active engagement into the learning and reasoning processes performed by the AI systems. This has become an increasingly important issue in view of the recent General Data Protection Regulation (GDPR) which requires actions taken as a result of a prediction from a learned model to be justified.

There has been a growing interest in logic-based machine learning approaches whose learned models are explainable and human interpretable. The goal of these approaches is the automated acquisition of knowledge (expressed as a logic program) from given (labelled) examples and existing background knowledge. One of the main advantage of these machine learning approaches is that the learned knowledge can be easily expressed into plain English and explained to a human user, so facilitating a closer interaction between humans and the machine. Logic-based machine learning has traditionally addressed the task of learning knowledge expressible in a very limited form [14] (definite clauses). Our logic-based machine learning systems [1, 2, 7] have extended this field to a wider class of formalisms for knowledge representation, captured by the answer set programming (ASP) semantics [4]. This ASP formalism is truly declarative, and due to its non-monotonicity it is particularly well suited to commonsense reasoning It allows constructs such as choice rules, hard and weak constraints, and support for default inference and default assumptions. Choice rules and weak constraints are particularly useful for modelling human preferences, as the choice rules can represent the choices available to the user, and the weak constraints can specify which choices a human prefers. In the recent years we have made fundamental contributions to the field of logic-based machine learning by extending it to the learning of the full class of ASP programs [5]. Early approaches to learning ASP programs can mostly be divided into two categories: *brave* learners aim to learn a program such that at least one answer set covers the examples; on the other hand, *cautious* learners aim to find a program which covers the examples in all answer sets. Most of the early

ASP-based ILP systems were brave. In [7], we showed that some ASP programs cannot be learned using either the brave or the cautious settings, and in fact a combination of *both* brave and cautious semantics is needed. This has the original motivation for the *Learning from Answer Sets* family of frameworks [8–11] which we have developed since then and have been shown to be able to learn any ASP program.

One of the main features of our LAS framework is the ability to support non-monotonic learning. Non-monotonicity permits incremental learning, allowing the machine to periodically revise rules and knowledge learnt, as examples of user behaviours are continuously observed. The non-monotonicity property is particularly relevant in pervasive computing, where systems are expected to autonomously adapt to changes in user context and behaviour, whilst operating seamlessly with minimal user intervention. We have used our non-monotonic LAS systems in mobile privacy [13], where devices learn and revise user's models from sensory input and user actions (e.g. user's actions on mobile devices), and in security [3], where anomaly detection policies are learned from historical data using domain-specific function for scoring candidate rules to guide the learning process towards the best policies. In both applications, the declarative representation of the learned programs make them explainable to human users, and providing way for users to understand and amend what has been learnt.

Often, many alternative solutions can be learned to explain given set of examples, and most logic-based learning systems employ a bias towards shorter solutions, based on Occam's razor (the solution with the fewest assumptions is the most likely). Choosing the shortest hypothesis may lead to very general hypotheses being learned from relatively few examples. While this can be a huge advantage of logic-based machine learning over other machine learning approaches that need larger quantities of data, learning such general rules without sufficient quantities of data to justify them may not be desirable in every application domain. For example, in access control, wrongly allowing access to a resource may be far more dangerous than wrongly denying access. So, learning a more general hypothesis, representing a more permissive policy, would be more dangerous than a specific hypothesis, representing a more conservative policy. Equally, for access control where the need for resources is time critical, wrongly denying access could be more dangerous than wrongly allowing access. When learning such policies, and choosing between alternative hypotheses, it would be useful to specify whether the search should be biased towards more or less general hypotheses. In [6], we have proposed a logic-based machine learning system, called FastLAS, targeted at solving a restricted version of the context-dependent learning from answer sets tasks that require only observational predicate learning. This system has two main advantages: it allows for domain-specific scoring function for hypotheses which generalises the standard Occam's razor approach, where hypotheses with the lowest number of literals are normally assumed to be preferred; and it is specifically designed to be scalable with respect to the hypothesis space. Its restriction to observational predicate learning has been lifted in [12], where the FastNonOPL system is proposed to solve non observational predicate learning from answer set tasks, whilst preserving scalability is a challenging open problem.

References

1. Athakravi, D., Corapi, D., Broda, K., Russo, A.: Learning through hypothesis refinement using answer set programming. In: Zaverucha, G., Santos Costa, V., Paes, A. (eds.) ILP 2013. LNCS, vol. 8812, pp. 31–46. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44923-3_3
2. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming in answer set programming. In: Muggleton, S.H., Tamaddoni-Nezhad, A., Lisi, F.A. (eds.) ILP 2011. LNCS, vol. 7207, pp. 91–97. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31951-8_12
3. Drozdov, A., Law, M., Lobo, J., Russo, A., Don, M.W.: Online symbolic learning of policies for explainable security. In: 3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2021, Atlanta, GA, USA, 13–15 December 2021, pp. 269–278 (2021)
4. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP, vol. 88, pp. 1070–1080 (1988)
5. Law, M.: Inductive learning of answer set programs. Ph.D. thesis, Imperial College London, UK (2018). https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.762179
6. Law, M., Russo, A., Bertino, E., Broda, K., Lobo, J.: Fastlas: scalable inductive logic programming incorporating domain-specific optimisation criteria. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020, pp. 2877–2885 (2020)
7. Law, M., Russo, A., Broda, K.: Inductive learning of answer set programs. In: Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, 24–26 September 2014, Proceedings, pp. 311–325 (2014)
8. Law, M., Russo, A., Broda, K.: Learning weak constraints in answer set programming. Theory Pract. Logic Program. **15**(4–5), 511–525 (2015)
9. Law, M., Russo, A., Broda, K.: Iterative learning of answer set programs from context-dependent examples. Theory Pract. Logic Program. (2016)
10. Law, M., Russo, A., Broda, K.: The complexity and generality of learning answer set programs. Artif. Intell. **259**, 110–146 (2018). https://doi.org/10.1016/j.artint.2018.03.005
11. Law, M., Russo, A., Broda, K.: The ILASP system for inductive learning of answer set programs. CoRR abs/2005.00904 (2020). https://arxiv.org/abs/2005.00904
12. Law, M., Russo, A., Broda, K., Bertino, E.: Scalable non-observational predicate learning in ASP. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event/Montreal, Canada, 19–27 August 2021, pp. 1936–1943 (2021)
13. Markitanis, A., Corapi, D., Russo, A., Lupu, E.: Learning user behaviours in real mobile domains. In: Latest Advances in Inductive Logic Programming. Imperial College Press (ILP 2011 Post-proceeding)
14. Muggleton, S.: Inverse entailment and progol. New Gener. Comput. **13**(3–4), 245–286 (1995)

Abstract Argumentation with Focus on Argument Claims – An Overview[1]

Michael Bernreiter, Wolfgang Dvořák, Anna Rapberger, and Stefan Woltran

TU Wien, Institute of Logic and Computation, Austria
{mbernrei, dvorak, arapberg, woltran}@dbai.tuwien.ac.at

Abstract. Abstract argumentation frameworks are among the best researched formalisms in the last two decades. They can be used to model discourses, provide a common ground for several nonmonotonic logics, and are employed to define semantics for more advanced argumentation formalisms. In the latter two domains, it is not the abstract argument's name, but the claim the argument represents, which should be in the focus of reasoning tasks. In this context, the fact that different arguments can represent the same claim leads to certain intricacies when it comes to the actual definition of semantics and in terms of computational aspects. In this talk, we give an overview on recent results in this direction. Those include the relation between argumentation and logic programming semantics, as well as a complexity analysis of acceptance problems in terms of claims and the effect of preferences in this setting.

Keywords: Argumentation Semantics · Claim-based Reasoning · Computational Complexity · Preferences

A Claim-Based Perspective on Logic Programming

Computational argumentation is a vibrant research area in AI [1, 2]; it is concerned with conflict resolution of inconsistent information and the justification of *defeasible statements (claims)* through logical or evidence-based reasoning. The *abstract* representation of conflicting information, significantly shaped by Dung [6], is among the most prominent approaches in this context. In his *abstract argumentation frameworks (AFs)*, each argument is treated as an abstract entity while an attack relation encodes (asymmetric) conflicts between them. Acceptance of arguments is evaluated with respect to *argumentation semantics*. In recent years, the acceptance of *claims* received increasing attention [3, 9]. *Claim-augmented argumentation frameworks (CAFs)* [9] extend Dung's model by assigning each argument its own claim, allowing for systematic study of structural and relational properties of claim acceptance. Formally, a CAF is a triple (A, R, cl) consisting of a set of arguments A, a set of directed attacks

[1] Supported by WWTF through project ICT19-065, and FWF through projects P30168 and W1255-N23.

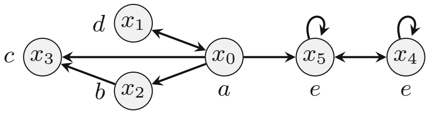
$R \subseteq A \times A$ between arguments, and a claim-function cl assigning a claim to each argument. They can be represented as directed labeled graphs (cf. Example 1).

Argumentation and logic programming are closely related [5, 6, 13]. The correspondence of stable model semantics with stable semantics in AFs is probably the most fundamental example [6], but also other logic programming semantics admit equivalent argumentation semantics [13]. With CAFs, the correspondence is particularly close: when identifying *atoms* in a given logic program (LP) P with *claims* of arguments constructed from rules in P we obtain a natural correspondence between LP semantics and AF semantics in terms of claims.

Example 1 (adapted from [5]). Consider the following logic program P:

$r_0 : a \leftarrow \texttt{not}\ d$	$r_2 : b \leftarrow \texttt{not}\ a$	$r_4 : e \leftarrow \texttt{not}\ e$
$r_1 : d \leftarrow \texttt{not}\ a$	$r_3 : c \leftarrow \texttt{not}\ a, \texttt{not}\ b$	$r_5 : e \leftarrow \texttt{not}\ a, \texttt{not}\ e$

When we interpret each rule r_i as an argument x_i with claim $head(r_i)$ and consider attacks between arguments x_i and x_j if the claim of x_i appears negated in the body of the rule r_j corresponding to x_j, we obtain the following CAF $\mathcal{F}$:



P returns $\emptyset$, $\{a\}$, and $\{d, b\}$ under *p-stable model semantics* (where one allows for undecided atoms). The complete[2] argument-sets of $\mathcal{F}$ in turn are $\emptyset$, $\{x_0\}$, and $\{x_1, x_2\}$. Inspecting the claims of these sets, $\mathcal{F}$ thus yields the same outcome as P. This is not a coincidence: as shown in [5], complete semantics correspond to p-stable model semantics when extracting the claims of the arguments. $\diamond$

Hence the representation as CAF establishes the connection between the two paradigms without detours, i.e., no additional steps or mappings are needed. Moreover, with CAFs, it is possible to capture semantics that make direct use of the claims. This advantage becomes apparent when we consider semantics that take *false* atoms into account: *L-stable semantics* [10] minimizes the set of undecided atoms in a p-stable model. *Semi-stable semantics* can be seen as their AF counter-part: here, the set of arguments which are neither accepted (i.e., contained in a complete extension) nor attacked is minimized. However, when evaluating our LP P under L-stable model semantics and its corresponding CAF $\mathcal{F}$ under semi-stable semantics we observe an undesired discrepancy.

Example 2 (Example 1 ctd.). The L-stable models of P are $\{a\}$ (atoms b, c, d are false) and $\{d, b\}$ (here, a, c are false). We obtain the single semi-stable extension $\{x_0\}$ in $\mathcal{F}$ (x_0 attacks all remaining arguments except x_4, minimizing undecided arguments), hence $\{a\}$ is the only semi-stable claim-set of $\mathcal{F}$. $\diamond$

[2] A set of arguments E is complete if it is conflict-free, defends itself, and contains all arguments it defends (E defends a if each attacker b of a is counter-attacked).

With CAFs, it is possible to define argumentation semantics such that they mimic the behavior of performing maximization on conclusion-level and minimize the set of undecided *claims* instead. For this, it is crucial to consider the *defeated claims* of an extension: intuitively, a claim is defeated iff all arguments carrying this claim are attacked. We illustrate the idea on our example.
Example 3 (Example 1 ctd.). We model L-stable semantics by maximizing accepted and defeated claims of complete sets: The set $\{x_0\}$ defeats claims b, c, d; claim e is not defeated because x_0 does not attack all occurrences of e. The set of accepted and defeated claims w.r.t. the extension $\{x_0\}$ (the *claim-range* of $\{x_0\}$) is thus given by $\{a, b, c, d\}$. Note that we obtain the same claim-range for the complete set $\{x_1, x_2\}$, which provides us with two semi-stable extensions under this evaluation method. Now, $\mathcal{F}$ yields the same outcome as P when evaluated under L-stable semantics. $\diamond$

Advances in Claim-Based Reasoning in a Nutshell

Claim-Based Semantics. We sketched two different evaluation methods for CAFs, taking into account different aspects of claim-based reasoning: In the first method, semantics are evaluated with respect to the underlying AF and the claims are extracted in the final step of the evaluation. We call this variant *inherited semantics* [9]. In the second method, we considered claim-defeat (as illustrated in Example 3) and performed maximization on claim-level. We call this variant *claim-level semantics* [8, 12]. Both variants capture claim-based reasoning in different aspects. While inherited semantics are well-suited to investigate justification in structured argumentation, claim-level semantics capture reasoning in conclusion-oriented formalisms. In the talk, we will review a detailed comparison between these two variants, cf. [8].

Well-formed CAFs. Observe that in CAFs obtained from LPs (cf. Example 5), any two arguments with the same claim attack the same arguments, i.e., if x and y have the same claim then x attacks the argument z iff y attacks z. This behavior is common to many instantiations of CAFs, and gives rise to the important class of well-formed CAFs. Formally, a CAF (A, R, cl) is well-formed iff for all $x, y \in A$ with $cl(x) = cl(y)$ we have $\{z \mid (x, z) \in R\} = \{z \mid (y, z) \in R\}$. As mentioned, well-formed CAFs capture LP-instantiations. Moreover, well-formed CAFs have benefits over general CAFs with regards to semantical properties and computational complexity. We furthermore note that the inherited and claim-based versions of prominent (e.g., stable) semantics coincide on well-formed CAFs [8].

Preferences in Claim-Based Reasoning. While well-formed CAFs are a natural sub-class of CAFs, they fail to account for a notion common to many formalisms instantiated into AFs, namely preferences. Specifically, in the course of the instantiation process, it often occurs that one argument x is considered stronger than (or: preferred to) another argument y ($x \succ y$). If there is an attack violating this preference, i.e., $(y, x) \in R$, then this is called a critical attack. This notion of preference in terms of argument strength leads to a generalization of well-formed CAFs to so-called Preference-based CAFs (PCAFs) [4]. Formally, a PCAF is given as $(A, R, cl, \succ)$ where (A, R, cl) is a well-formed CAF and $\succ$ is an asymmetric preference relation over A.

Preferences are then resolved via so-called preference-reductions which transform PCAFs into CAFs. The literature [11] describes several such reductions for AFs: a prominent method is to delete critical attacks; further approaches revert critical attacks or delete them only if there is also an attack from the stronger to the weaker argument, Finally a combination of the latter two is often considered. These four reductions give rise to four new CAF-classes being strictly located between well-formed CAFs and general CAFs. Also, only some of these classes preserve certain benefits of well-formed CAFs while others exhibit the same behavior as general CAFs.

Complexity Results. In the talk, we finally review complexity results obtained for CAFs and PCAFs [4, 7, 9]. It has been shown that the verification problem (testing whether a given claim set is an extension for a given CAF/PCAF) can have higher complexity for CAFs than for AFs, while this gap does not show up for most semantics when restricting ourselves to well-formed CAFs. Interestingly, for PCAFs this effect depends on the chosen reduction.

References

1. Atkinson, K., et al.: Towards artificial argumentation. AI Mag. **38**(3), 25–36 (2017)
2. Baroni, P., Gabbay, D.M., Giacomin, M.: Handbook of Formal Argumentation. College Publications (2018)
3. Baroni, P., Riveret, R.: Enclling systems. J. Artif. Intell. Res. **66**, 793–860 (2019)
4. Bernreiter, M., Dvor˘ák, W., Rapberger, A., Woltran, S.: The effect of preferences in abstract argumentation under a claim-centric view. CoRR (2022). https://doi.org/10.48550/arXiv.2204.13305, accepted for publication at NMR'22
5. Caminada, M., Sá, S., Alcântara, J., Dvořák, W.: On the equivalence between logic programming semantics and argumentation semantics. Int. J. Approx. Reas. **58**, 87–111 (2015)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–358 (1995)
7. Dvořák, W., Greßler, A., Rapberger, A., Woltran, S.: The complexity landscape of claim-augmented argumentation frameworks. In: AAAI, pp. 6296–6303 (2021)
8. Dvořák, W., Rapberger, A., Woltran, S.: Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to SETAFs. In: KR, pp. 341–350 (2020)
9. Dvořák, W., Woltran, S.: Complexity of abstract argumentation under a claim-centric view. Artif. Intell. **285**, 103290 (2020)
10. Eiter, T., Leone, N., Saccà, D.: On the partial semantics for disjunctive deductive databases. Ann. Math. Artif. Intell. **19**(1–2), 59–96 (1997). https://doi.org/10.1023/A:1018947420290
11. Kaci, S., van der Torre, L.W.N., Vesic, S., Villata, S.: Preference in abstract argumentation. In: Handbook of Formal Argumentation, vol. 2, pp. 211–248. College Publications (2021)
12. Rapberger, A.: Defining argumentation semantics under a claim-centric view. In: STAIRS-ECAI. CEUR Workshop Proceedings, vol. 2655. CEUR-WS.org (2020)
13. Wu, Y., Caminada, M., Gabbay, D.M.: Complete extensions in argumentation coincide with 3-valued stable models in logic programming. Studia Logica **93**(2–3), 383–403 (2009)

Contents

Technical Contributions