Quantitative and Stream Extensions of Answer Set Programming

Rafael Kiesel
TU Vienna, Austria
rafael.kiesel@web.de

Answer Set Programming has separately been extended with constraints, to the streaming domain, and with capabilities to reason over the quantities associated with answer sets. We propose the introduction and analysis of a general framework that incorporates all three directions of extension by exploiting the strengths of Here-and-There Logic and Weighted Logic.

1 Introduction

While propositional Answer Set Programming (ASP) is already NP-hard and therefore powerful enough to express many challenging problems, their specification can be tedious and complicated. Further, there are relevant problems that require higher expressivity or reasoning over data that changes with time. This and the practical usage of ASP gave rise to a need for a simpler, more expressive, and more concise specification language [1, 11]. Thus, ASP was extended in multiple directions. We focus on the following ones:

1. Time Domain (TD): In [5] ASP-semantics were combined with a temporal context resulting in the Logic-based framework for Analytic Reasoning over Streams (LARS). Here, interpretations assign possibly different sets of facts to time points. Accordingly, the input language was extended with operators like $\Diamond$, corresponding to existential quantification over time points. Another temporal extension of ASP is Temporal Equilibrium Logic (TEL) [9].
2. Quantitative Reasoning over Models (QM): Given a program we may not only be interested in its answer sets but also in reasoning with quantities associated with them. Commonly this includes:
   2.1 Probabilities of Models [2, 21, 28, 33]
   2.2 Preferences over Models [21, 7]
   2.3 Weighted Model Counting [20]
3. Quantitative Constraints (QC): In order to simplify the specification language of ASP and improving its succinctness different constraints were introduced as basic language elements, among them:
   3.1 Aggregates [17, 11]
   3.2 Weight Constraints [29]
   3.3 Arithmetic Operations [22]
   3.4 Guessing in Rule-Heads [29, 22]

This significantly enhanced the expressive power of programs and facilitated their specification. Naturally, we would like to use combinations of these directions. A current area that would benefit highly is that of traffic routing: Decisions need to be made based on dynamic temporal information (→ TD), quantitative information about the traffic (→ QC) and these decisions should be optimized in such a way that the waiting times are small (→ QM).

A. Formisano, Y.A Liu, et al. (Eds.): International Conference on
Logic Programming (Technical Communications) 2021 (ICLP 2021)
EPTCS 345, 2021, pp. 267–275, doi:10.4204/EPTCS.345.43

2 Problem Statement

We are interested in a general framework which combines TD, QM, and QC, that is, we want to

Find and analyze a general framework that allows for succinct specifications and reasoning over answer sets in a streaming context.

When *finding* such a general framework we are faced with the following two main challenges.

First, we have to decide how to lift the quantitative features to the temporal domain *in a structured way*. There are many extensions introducing QM or QC and while quantitative extensions within a category share a purpose they tackle different *sub*-problems. Furthermore, there are sometimes even multiple approaches for just one sub-problem that are defined independently. Already their relation and especially their combination is not always discussed, or only in post-hoc analysis [21]. It follows that we need to find a reasonable way to capture the *combined* capabilities introduced by the different extensions in a preferably *uniform* manner. Reasonable here meaning a strategy different from simply taking an unstructured "union" of the different extensions. On the other hand, we need to avoid restricting ourselves to a significantly less powerful fragment.

Second, we need to incorporate the temporal domain. For this purpose, one can not simply reuse previous definitions in a straight forward way. To illustrate this, consider an aggregate expression of the form $\texttt{sum}\{X : p(X)\}$ over a temporal domain, where $p(x)$ may hold at some time point but not at a different one. Here, we need to be able to specify whether the sum should consider only the values x such that $p(x)$ holds at the current time point or those for which there exists a time point such that $p(x)$ holds. Furthermore, in the latter case, we need to differentiate whether the same value should be counted only once or according to its multiplicity. Similar problems occur also with other capabilities.

Given that we find an appropriate framework, we further need to *analyze* it, in order to make it practically useful. An implementation and its application to real world use cases require an investigation of efficiently solvable and safe fragments.

Detailed research questions concerning the creation and analysis of such a framework are given after a discussion of our methodology.

3 Approach

The QC and QM extensions are mostly of a *quantitative* nature, i.e. they involve some form of calculation. In order to capture possible computations abstractly, in an algebraic manner, we consider semirings.

A semiring $\mathscr{R}$ is an algebraic structure of the form $\mathscr{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ where $\oplus$ and $\otimes$ are addition and multiplication with neutral elements $e_\oplus$ and $e_\otimes$ respectively. Using semirings one can capture various modes of calculation ranging from the the tropical semiring $\mathscr{R}_{\texttt{trop}} = ([0,\infty], \min, +, \infty, 0)$ to the well known semiring over the real numbers $\mathbb{R} = (\mathbb{R}, +, \cdot, 0, 1)$. Semiring have previously been successfully used to capture different semantics in a uniform syntax, by parameterising definitions with a semiring. Bistarelli et al. [6] used them to define constraint semantics and Kimming et al. [20] provided a rule language with parameterised calculation.

Notably, *Weighted Logic* introduced by Droste and Gastin [12] connects calculations in semirings and logical formulas. So called *weighted formulas* like

$$\alpha = 15 \wedge \texttt{Circus} \vee 20 \wedge \texttt{Restaurant}$$

can contain weights from a semiring $\mathscr{R}$ in addition to logical formulas. This allows the specification of calculations, which depend on the truth of logical formulas, by interpreting disjunctive connectives as addition $\oplus$ and conjunctive connectives as multiplication $\otimes$. In line with this, the neutral elements $e_\oplus$ and $e_\otimes$ replace falsum ($\bot$) and truth ($\top$). We slightly deviate from Droste and Gastin's original definition and use the more intuitive notation for the same formula

$$\alpha = 15 * \texttt{Circus} + 20 * \texttt{Restaurant}$$

Over the semiring $\mathbb{R}$ the weighted formula α encodes the amount of money spent on an evening depending on which places are visited. When `Circus` is false and `Restaurant` is true, its semantics is $15 \cdot 0 + 20 \cdot 1 = 20$. Over the tropical semiring $\mathscr{R}_{\texttt{trop}}$ it encodes the price of the cheapest possible activity. In the same situation its semantics is $\min(15+\infty, 20+0) = 20$.

For QM it is sufficient to use the classical weighted semantics. When it comes to QC it is necessary to define answer set semantics for weighted formulas, since they are part of the program and influence which atomic formulas are asserted in answer sets.

Since are often hard to For this, we employ Here-and-There (HT) Logic [31]. It can be seen as a fragment of intuitionistic logic with two worlds Here and There, with one interpretation $\mathscr{I}_H$ and $\mathscr{I}_T$ each such that when an atom holds Here then it also holds There. The semantics satisfies that given a program Π an HT-interpretation $\langle \mathscr{I}_H, \mathscr{I}_T \rangle$ satisfies Π if $\mathscr{I}_T$ classically satisfies the reduct of Π with respect to $\mathscr{I}_H$. Then the answer sets $\mathscr{I}$ of Π simply correspond to those interpretations $\mathscr{I}$ such that $\langle \mathscr{I}, \mathscr{I} \rangle$ satisfies Π but for any subset $\mathscr{I}' \subsetneq \mathscr{I}$ it holds that $\langle \mathscr{I}', \mathscr{I} \rangle$ does not satisfy Π.

It was shown that this definition of answer sets is equivalent to the typical reduct-based definitions [24, 19] and can be combined well with other logics like first-order logic [32] or temporal logic [9] giving them an answer set semantics.

4 Research Plan & Progress

Weighted Logic combines Boolean logic and algebraic expressions leading to the possibility to specify calculations dependent on the satisfaction of logical formulas with ease. In Weighted HT Logic, a combination of Weighted Logic and HT Logic, this dependence follows the intuition of answer set semantics. On the one hand, Weighted Logic allows many different modes of calculation due to the variability of the underlying semiring. On the other hand, its syntax and semantics is homogeneous for each of those modes. This makes Weighted (HT) Logic a promising approach to formalize many quantitative extensions of ASP for QM and QC using a common homogeneous language.

Furthermore, Weighted Logic is rather generic: for many two-valued logics it is possible to define a weighted version, by using the intuition that conjunction is multiplication and disjunction is addition. This facilitates the combination with the temporal domain.

We aim to find a general framework using Weighted (HT) Logic. Thus, we consider the following research questions:

Generality & Faithfulness

(Q1) Can we use Weighted Logic for General Quantitative Reasoning over Models?

(Q2) Can we use Weighted HT Logic for General Quantitative Constraints?

(Q3) Can we capture temporal aspects using Weighted (HT) Logic?

The first concern here is *Generality*, i.e. whether the capabilities of previous extensions are subsumed to a reasonable degree. The second is *Faithfulness*, i.e. if the capabilities in our framework correctly capture the intended semantics. Q1 and Q2 focus on QM and QC respectively, whereas Q3 asks whether lifting the combination of ASP and Weighted (HT) Logic to LARS/TEL sufficiently captures the temporal aspects of TD.

Q1 has been answered positively in [15]. We considered Weighted LARS (wLARS) formulas as a means of extracting quantities from answer streams of LARS programs. This lead to a general framework for the QM extensions. wLARS formulas can be used for probabilistic reasoning, preferential reasoning and weighted model counting, respectively via normalization, optimization and aggregation over all answer streams of a program. We showed how three prominent extensions of logic programming with QM capabilities of different natures, namely P-log [2], (a)Problog [33, 20] and LP^{MLN} [21] can be faithfully expressed using wLARS. Apart from that, we considered preferential reasoning with wLARS in more detail and demonstrated that the temporal aspects that need to be considered when obtaining quantities are also covered by wLARS. This is a first step towards answering Q3 positively.

Secondly, Q2 has been answered positively in [14]. In order to achieve a similar level of generality for QC, we defined First-Order Weighted HT Logic. Here, the "first-order quantifiers" naturally correspond to aggregation. Weighted formulas were used to define *algebraic constraints* $k \sim_{\mathscr{R}} \alpha$, which are (in)equations between a value k and a weighted formula α over the semiring $\mathscr{R}$. They can be used to assert (in rule-heads) or check (in rule-bodies) whether an interpretation satisfies a restriction on some quantity (specified by α). Taking our previous example formula α, the algebraic constraint $30 >_{\mathbb{R}} 15 * \texttt{Circus} + 20 * \texttt{Restaurant}$ is satisfied whenever a budget of 30 is sufficient to pay for the places visited.

We found that all the identified capabilities - aggregation, weight constraints, arithmetic operations, ... - are faithfully incorporated in our framework, with only mild practical restrictions. We even show that there are some new capabilities, like so called ”minimized choice constraints”, that have not been considered before.

Therefore, the question of generality and faithfulness has already been answered positively. Only the lifting along TD using LARS may cause a problem. While there is an HT Logic for LARS, it leads to a different semantics. It remains to be seen whether this can be fixed. Otherwise, TEL, which is defined using HT Logic and adds operators for reasoning over streams, is a promising alternative.

Theoretical Analysis

(Q4) Which properties of programs carry over and which new ones can be found?

(Q5) What does the complexity of reasoning depend on?

The following properties are known to be of interest:

- **Safety**, a property necessary for ASP with variables. Considered, inter alia, in [18, 25, 10]
- **Finite Groundability**, a stronger restriction than safety that entails that programs can be evaluated over a finite domain. Considered, inter alia, in [23, 13, 3]
- **Modularity**, a property facilitating the parallel evaluation of answer set programs. Considered, inter alia, in [27, 17]
- **Program Equivalence**, a property used in the simplification of answer set programs. Considered, inter alia, in [30, 26, 4]

Safety and finite groundability are essential for practical usability. Unsafe programs may have unintuitive semantics. Finite Groundability ensures that one can reduce reasoning tasks for programs with variables to the corresponding reasoning tasks for programs without variables.

We already introduced safety conditions for programs with algebraic constraints [14]. Furthermore, we combined and transferred the concepts of domain restrictedness from [29] and argument restrictedness from [23] to our framework, resulting in a class of finitely ground programs, for which membership can be decided tractably. Also, regarding program equivalence, we showed that two programs with algebraic constraints are strongly equivalent iff their HT-models coincide [14].

Modularity may still be considered in-depth later on.

As for complexity, our preliminary results show strong influence by two parameters. The first is which semirings are allowed: Depending on the semiring a single addition can be incomputable or constant. Already in [15, 14] we gave a preliminary consideration of the complexity of the respective introduced reasoning tasks. There, we identified some sufficient conditions on semirings for efficient calculations, which allowed us to provide upperbounds on the complexity.

However, we noticed that even under the chosen conditions the choice of semiring leads to a lot of variation in the complexity: In the context of QM the problem of Algebraic Answer Set Counting (AASC) (i.e. counting answer sets, which have a weight over a semiring) can be NP, #P or OPTP-complete depending on the choice of the semiring. This motivated a more in depth consideration of the class of Sum-Of-Products problems over semirings in [16]. Apart from AASC, also many other problems (viz. [20, 6, 14]) fall into this class of problems. We introduced NP($\mathscr{R}$) a generalization of NP parameterized with a semiring $\mathscr{R}$ together with a prototypical complete problem SAT($\mathscr{R}$) and showed that AASC and many other problems are NP($\mathscr{R}$)-complete. Furthermore, we analyzed the relation of NP($\mathscr{R}$) to classical complexity classes in dependence on the semiring $\mathscr{R}$ and the properties satisfied by it.

The second parameter is the allowed language fragment. We can tune both the expressiveness and the restrictions on variable occurrences. We plan to consider different conditions for finite groundability and consider the complexity entailed by using them.

Empirical Analysis

(Q6) To what extent can we efficiently implement our framework?

(Q7) How can our framework be employed in real world applications?

We obtained promising preliminary results regarding Q6 by considering AASC, i.e., Weighted LARS restricted to non-disjunctive ASP [34]. Here, we introduced a new way to translate a program Π into propositional formula ϕ such that the treewidth of ϕ is bounded by kc, where k is the treewidth of Π and c is *component-boosted backdoor size*, a novel parameter that measures the cyclicity of the dependency graph of Π. In addition to that, our translation satisfies that the (weighted/algebraic) answer set count of Π is equal to the (weighted/algebraic) model count of ϕ. Our experimental results showed that our prototype implementation aspmc[1] outperforms the current standard tool called Problog [33] on a standard benchmark set from [35]. We thus argue that aspmc provides an efficient implementation for QM. It remains to be seen in the future, how this implementation can be extended to the temporal domain, i.e., Weighted LARS.

For QC an implementation is further in the future and it seems likely that we will focus on extending aspmc to also handle the temporal domain. In the same step, we also plan to apply it in a real world application in the context of real time traffic regulation in order to answer Q7.

[1] available at https://github.com/hmarkus/asp2sat (open source).

5 State of the Art

To the best of my knowledge, the idea of using Weighted Logic as a means of adding general quantitative reasoning capabilities to ASP is novel. There has however already been research useful for expressing multiple quantitative extensions of ASP in a common terminology. In the following, we go over different quantitative extensions of ASP aiming at generality.

Hybrid ASP [8] The authors defined an extension of HT Logic that includes general constraints and multi-valued interpretations providing a non-monotonic integration of constraint processing into ASP. Since the approach allows arbitrary constraints it captures QC extensions; however, as a consequence, the syntax of constraints (except over the reals) is left open, and specific constraint expressions like aggregates rely on extra definitions.

Nested Expressions [17] The formalism allows the nesting of constraints on aggregates. The only restriction on aggregates is that they have to be over the reals. Three QC extensions are partly subsumed by the approach.

LP$^{\mathrm{MLN}}$ [21] Lee and Yang's extension of logic programming with probabilistic reasoning has been shown to subsume two other QM extensions, namely ASP with weak constraints and P-log. This shows that LP$^{\mathrm{MLN}}$captures multiple capabilities along QM. It however neglects problems that are not over the reals.

Problog [33] Problog is an implementation for probabilistic logic programming. It even also allows *algebraic* answer set counting, however, it only allows a restricted fragment of ASP in the input language.

telingo [9] Telingo is a solver from the potassco family, built for a fragment of TEL, a logic for temporal reasoning under ASP semantics. It already combines TD and QC, however the temporal aspects are not fully incorporated in the quantitative constraints. It may be useful as a basis of an implementation for our work.

Acknowledgments

The author would like to thank Thomas Eiter for supervising his PhD research activities. This work has been supported by FWF project W1255-N23.

References

[1] Mario Alviano & Wolfgang Faber (2018): *Aggregates in Answer Set Programming*. *Künstliche Intell.* 32(2-3), pp. 119–124, doi:10.1007/s13218-018-0545-9.

[2] Chitta Baral, Michael Gelfond & J. Nelson Rushton (2009): *Probabilistic reasoning with answer sets*. *Theory Pract. Log. Program.* 9(1), pp. 57–144, doi:10.1017/S1471068408003645.

[3] Michael Bartholomew & Joohyung Lee (2010): *A Decidable Class of Groundable Formulas in the General Theory of Stable Models*. In Fangzhen Lin, Ulrike Sattler & Miroslaw Truszczynski, editors: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, AAAI Press, pp. 477–485. Available at `http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1375`.

[4] Harald Beck, Minh Dao-Tran & Thomas Eiter (2016): *Equivalent Stream Reasoning Programs*. In Subbarao Kambhampati, editor: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, IJCAI/AAAI Press, pp. 929–935. Available at `http://www.ijcai.org/Abstract/16/136`.

[5] Harald Beck, Minh Dao-Tran & Thomas Eiter (2018): *LARS: A Logic-based framework for Analytic Reasoning over Streams*. *Artif. Intell.* 261, pp. 16–70, doi:10.1016/j.artint.2018.04.003.

[6] Stefano Bistarelli, Ugo Montanari & Francesca Rossi (1997): *Semiring-based Constraint Logic Programming*. In: *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, Morgan Kaufmann, pp. 352–357. Available at `http://ijcai.org/Proceedings/97-1/Papers/055.pdf`.

[7] Francesco Buccafurri, Nicola Leone & Pasquale Rullo (1997): *Strong and Weak Constraints in Disjunctive Datalog*. In Jürgen Dix, Ulrich Furbach & Anil Nerode, editors: *Logic Programming and Nonmonotonic Reasoning, 4th International Conference, LPNMR'97, Dagstuhl Castle, Germany, July 28-31, 1997, Proceedings*, *Lecture Notes in Computer Science* 1265, Springer, pp. 2–17, doi:10.1007/3-540-63255-7_2.

[8] Pedro Cabalar, Jorge Fandinno, Torsten Schaub & Philipp Wanko (2020): *A Uniform Treatment of Aggregates and Constraints in Hybrid ASP*. In Diego Calvanese, Esra Erdem & Michael Thielscher, editors: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pp. 193–202, doi:10.24963/kr.2020/20.

[9] Pedro Cabalar, Roland Kaminski, Torsten Schaub & Anna Schuhmann (2018): *Temporal Answer Set Programming on Finite Traces*. *Theory Pract. Log. Program.* 18(3-4), pp. 406–420, doi:10.1017/S1471068418000297.

[10] Pedro Cabalar, David Pearce & Agustín Valverde (2009): *A Revised Concept of Safety for General Answer Set Programs*. In: *Logic Programming and Nonmonotonic Reasoning, 10th International Conference, LPNMR 2009, Potsdam, Germany, September 14-18, 2009. Proceedings*, pp. 58–70, doi:10.1007/978-3-642-04238-6_8.

[11] Tina Dell'Armi, Wolfgang Faber, Giuseppe Ielpa, Nicola Leone & Gerald Pfeifer (2003): *Aggregate Functions in Disjunctive Logic Programming: Semantics, Complexity, and Implementation in DLV*. In Georg Gottlob & Toby Walsh, editors: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, Morgan Kaufmann, pp. 847–852. Available at `http://ijcai.org/Proceedings/03/Papers/122.pdf`.

[12] Manfred Droste & Paul Gastin (2007): *Weighted automata and weighted logics*. *Theor. Comput. Sci.* 380(1-2), pp. 69–86, doi:10.1016/j.tcs.2007.02.055.

[13] Thomas Eiter, Michael Fink, Thomas Krennwallner & Christoph Redl (2013): *Liberal Safety Criteria for HEX-Programs*. In Marie desJardins & Michael Littman, editors: *Twenty-Seventh AAAI Conference (AAAI 2013), July 14–18, 2013, Bellevue, Washington, USA*, AAAI Press, pp. 267–275. Available at `http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6209`.

[14] Thomas Eiter & Rafael Kiesel (2020): *ASP(AC): Answer Set Programming with Algebraic Constraints*. *Theory Pract. Log. Program.* 20(6), pp. 895–910, doi:10.1017/S1471068420000393.

[15] Thomas Eiter & Rafael Kiesel (2020): *Weighted LARS for Quantitative Stream Reasoning*. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín & Jérôme Lang, editors: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, *Frontiers in Artificial Intelligence and Applications* 325, IOS Press, pp. 729–736, doi:10.3233/FAIA200160.

[16] Thomas Eiter & Rafael Kiesel (2021): *On the Complexity of Sum-of-Products Problems over Semirings*. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, pp. 6304–6311. Available at `https://ojs.aaai.org/index.php/AAAI/article/view/16783`.

[17] Paolo Ferraris (2011): *Logic programs with propositional connectives and aggregates*. *ACM Trans. Comput. Log.* 12(4), pp. 25:1–25:40, doi:10.1145/1970398.1970401.

[18] Martin Gebser, Amelia Harrison, Roland Kaminski, Vladimir Lifschitz & Torsten Schaub (2015): *Abstract gringo*. *TPLP* 15(4-5), pp. 449–463, doi:10.1017/S1471068415000150.

[19] Michael Gelfond & Yuanlin Zhang (2014): *Vicious Circle Principle and Logic Programs with Aggregates*. *Theory Pract. Log. Program.* 14(4-5), pp. 587–601, doi:10.1017/S1471068414000222.

[20] Angelika Kimmig, Guy Van den Broeck & Luc De Raedt (2011): *An Algebraic Prolog for Reasoning about Possible Worlds*. In Wolfram Burgard & Dan Roth, editors: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, AAAI Press, pp. 209–214. Available at `http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3685`.

[21] Joohyung Lee & Zhun Yang (2017): *LPMLN, Weak Constraints, and P-log*. In Satinder P. Singh & Shaul Markovitch, editors: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, AAAI Press, pp. 1170–1177. Available at `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14547`.

[22] Yuliya Lierler (2014): *Relating constraint answer set programming languages and algorithms*. *Artif. Intell.* 207, pp. 1–22, doi:10.1016/j.artint.2013.10.004.

[23] Yuliya Lierler & Vladimir Lifschitz (2009): *One More Decidable Class of Finitely Ground Programs*. In Patricia M. Hill & David Scott Warren, editors: *Logic Programming, 25th International Conference, ICLP 2009, Pasadena, CA, USA, July 14-17, 2009. Proceedings*, *Lecture Notes in Computer Science* 5649, Springer, pp. 489–493, doi:10.1007/978-3-642-02846-5_40.

[24] Vladimir Lifschitz (2008): *What Is Answer Set Programming?* In Dieter Fox & Carla P. Gomes, editors: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, AAAI Press, pp. 1594–1597. Available at `http://www.aaai.org/Library/AAAI/2008/aaai08-270.php`.

[25] Vladimir Lifschitz (2016): *Intelligent Instantiation and Supersafe Rules*. In: *Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, October 16-21, 2016, New York City, USA*, pp. 7:1–7:14, doi:10.4230/OASIcs.ICLP.2016.7.

[26] Vladimir Lifschitz, David Pearce & Agustín Valverde (2001): *Strongly equivalent logic programs*. *ACM Trans. Comput. Log.* 2(4), pp. 526–541, doi:10.1145/383779.383783.

[27] Vladimir Lifschitz & Hudson Turner (1994): *Splitting a Logic Program*. In Pascal Van Hentenryck, editor: *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, MIT Press, pp. 23–37.

[28] Matthias Nickles & Alessandra Mileo (2015): *A System for Probabilistic Inductive Answer Set Programming*. In Christoph Beierle & Alex Dekhtyar, editors: *Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, *Lecture Notes in Computer Science* 9310, Springer, pp. 99–105, doi:10.1007/978-3-319-23540-0_7.

[29] Ilkka Niemelä, Patrik Simons & Timo Soininen (1999): *Stable Model Semantics of Weight Constraint Rules*. In Michael Gelfond, Nicola Leone & Gerald Pfeifer, editors: *Logic Programming and Nonmonotonic Reasoning, 5th International Conference, LPNMR'99, El Paso, Texas, USA, December 2-4, 1999, Proceedings*, *Lecture Notes in Computer Science* 1730, Springer, pp. 317–331, doi:10.1007/3-540-46767-X_23.

[30] Juan Carlos Nieves & Helena Lindgren (2015): *Possibilistic nested logic programs and strong equivalence*. *Int. J. Approx. Reasoning* 59, pp. 1–19, doi:10.1016/j.ijar.2015.01.004.

[31] David Pearce & Agustín Valverde (2004): *Towards a First Order Equilibrium Logic for Nonmonotonic Reasoning*. In José Júlio Alferes & João Alexandre Leite, editors: *Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, Lisbon, Portugal, September 27-30, 2004, Proceedings*, *Lecture Notes in Computer Science* 3229, Springer, pp. 147–160, doi:10.1007/978-3-540-30227-8_15.

[32] David Pearce & Agustín Valverde (2008): *Quantified Equilibrium Logic and Foundations for Answer Set Programs*. In Maria Garcia de la Banda & Enrico Pontelli, editors: *Proc. ICLP'08*, Springer, pp. 546–560, doi:10.1007/978-3-540-89982-2_46.

[33] Luc De Raedt, Angelika Kimmig & Hannu Toivonen (2007): *ProbLog: A Probabilistic Prolog and Its Application in Link Discovery*. In Manuela M. Veloso, editor: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 2462–2467. Available at http://ijcai.org/Proceedings/07/Papers/396.pdf.

[34] Markus Hecher Thomas Eiter & Rafael Kiesel (2021 (to appear)): *Treewidth-aware Cycle Breaking for Algebraic Answer Set Counting*. In: *Eighteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

[35] Efthymia Tsamoura, Víctor Gutiérrez-Basulto & Angelika Kimmig (2020): *Beyond the Grounding Bottleneck: Datalog Techniques for Inference in Probabilistic Logic Programs*. 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, February 7-12, 2020*, AAAI Press, pp. 10284–10291. Available at https://aaai.org/ojs/index.php/AAAI/article/view/6591.