A Unifying Framework for Semiring-Based Constraint Logic Programming With Negation

Jeroen P. Spaans[1], Jesse Heyninck[1,2]

[1]Open Universiteit, The Netherlands
[2]University of Cape Town, South Africa

Abstract

Extended abstract of a paper published at IJCAI 2025[1] [2]—Constraint Logic Programming (CLP) is a logic programming formalism used to solve problems requiring the consideration of constraints, like resource allocation and automated planning and scheduling. It has previously been extended in various directions, for example to support fuzzy constraint satisfaction, uncertainty, or negation, with different notions of semiring being used as a unifying abstraction for these generalizations. None of these extensions have studied clauses with negation allowed in the body. We investigate an extension of CLP which unifies many of these extensions and allows negation in the body. We provide semantics for such programs, using the framework of approximation fixpoint theory (AFT), and give a detailed overview of the impacts of properties of the semirings on the resulting semantics. As such, we provide a unifying framework that captures existing approaches and allows extending them with a more expressive language.

Keywords

semiring-based constraint logic programming, negation, approximation fixpoint theory

Many problems require the consideration of constraints; examples of such problems can be relatively simple, like sudokus and similar puzzles, or more complex, like the real-world applications of resource allocation and automated planning and scheduling. Classically, a problem as such may be formulated as a Constraint Satisfaction Problem (CSP) and solved using Constraint Logic Programming (CLP). We investigate an extension of CLP capable of handling semiring-based constraints with negation.

Consider an informal example constraint logic program, which describes the allocation of a limited number of working hours to two different tasks.

```
% We define two tasks,
% taking 6 and 4 hours to complete respectively.
task(t1, 6).
task(t2, 4).
% A task is completed if enough time is scheduled.
completed(Task, HoursScheduled) :-
   task(Task, HoursRequired),
   HoursRequired ≤ HoursScheduled.
% We set a time limit of eight hours.
inTimeLimit(Hours1, Hours2) :-
   Hours1 + Hours2 ≤ 8.
% A schedule is evaluated by the degree to which both tasks
% are completed and the total allotted time does not exceed
% the time available.
schedule(HoursTask1, HoursTask2) :-
   completed(t1, HoursTask1),
   completed(t2, HoursTask2),
   inTimeLimit(HoursTask1, HoursTask2).
```

[1]A full version of the paper, including proofs and appendices, is available on arXiv [1].

23rd International Workshop on Nonmonotonic Reasoning, November 11–13, 2025, Melbourne, Australia

jeroen.spaans@ou.nl (J. P. Spaans); jesse.heyninck@ou.nl (J. Heyninck)

Using classical answer set programming semantics to evaluate this program with the goal `schedule(HoursTask1, HoursTask2)` returns *false*. Intuitively, this happens because no schedule can complete the two tasks—totalling ten hours of work—in less than eight hours. Knowing that the two tasks cannot both be completed in the time available, we may wish to optimize their partial completion instead. One way to do this is to replace *false* and *true* with values in $[0, 1]$ (where 1 represents complete truth and 0 complete falsity), replace *or* with $\max$, replace *and* with $\min$, and replace `HoursRequired` $\leq$ `HoursScheduled` with `HoursScheduled / HoursRequired`.

Our example demonstrated that CLP is limited to strict satisfaction or violation of constraints, and that we require an alteration of this framework to solve problems of constraint optimization. The same holds—for example—for problems involving fuzziness, uncertainty, or probability.

Bistarelli et al. [3, 4, 5] proposed Semiring-based Constraint Logic Programming, a generalization of CLP replacing the Boolean evaluation domain and the associated logical *and* and *or* connectives with semirings—algebraic structures consisting of a set equipped with an additive operator for disjunction and a multiplicative operator for conjunction—much like we did in our example. Since, many related formalisms have likewise been extended to the semiring setting [6, 7, 8, 9, 10, 11, 12] to certain success. Each of these works makes some assumptions about the semirings used, but what exactly those assumptions are and how they relate is left implicit or has not been studied. Herein lies the first major contribution of this work; we perform an analysis of the various families of semirings in relation to semiring-based semantics for CLP, paying special attention to the orderings each family gives rise to.

While some of the above-mentioned works permit negation, most do not, and a general analysis of negation in the semiring setting is so far absent from the literature. Herein lies the second major contribution of this work; a semiring-agnostic form of negation—based on negation in Gödel logics [13], and also used in the semiring-based formalism of Eiter and Kiesel [8]—is proposed and the effects of its addition on the semantics of semiring-based constraint logic programming are studied. The addition of negation gives us the expressive power to, for example, make the completion of a task in our scheduling problem contingent on the absence of blocking factors by replacing lines 5 through 8 of our example program with the following.

```
% A task is completed if enough time is scheduled
% and its completion is not blocked.
completed(Task, HoursScheduled) :-
   task(Task, HoursRequired),
   HoursRequired ≤ HoursScheduled,
   not blocked(Task).
```

Notably, and as is to be expected, the addition of negation leads to nonmonotonicity of the immediate consequence operator. To work around this problem we capture the new negation-permitting formalism in Approximation Fixpoint Theory (ADT) [14], endowing it with AFT's various semantics like Kripke-Kleene, Well-founded, and Stable, which generalize the semantics of normal logic programs.

Concretely, the contributions of this work come in five parts:

1. A novel notion of model—considering all contributing clauses at once, and specific to the semiring-based setting—is introduced and compared to the traditional notion of model which considers each clause's satisfaction separately.
2. The minimal model semantics based on these notions of model are then compared with the least fixpoint semantics based on an immediate consequence operator.
3. A generalized method for deriving orderings of semiring elements needed to define models and least fixpoints—but also needed in the later application of approximation fixpoint theory—is investigated and found to be a generalization of the method studied by Bistarelli et al. [5].
4. The application of a generalized notion of negation appearing at various points in the literature to our semiring-based framework is studied.
5. Approximation fixpoint theory is applied to our immediate consequence operator—made non-monotonic by the addition of negation—to define Kripke-Kleene, and well-founded and other stable semantics, studying both ultimate approximation and a novel approximator.

This work studies semiring-based semantics for constraint logic programming with negation, generalizing the approaches of [5] and [15]. Computational complexity, implementations, and applying AFT-based notions such as stratification [16], conditional independence [17] and non-determinism [18] are left as future work.

Acknowledgments

Acknowledgments This work was partially supported by the project LogicLM: Combining Logic Programs with Language Model with file number NGF.1609.241.010 of the research programme NGF AiNed XS Europa 2024-1, is (partly) financed by the Dutch Research Council (NWO).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J. Spaans, J. Heyninck, A Unifying Framework for Semiring-Based Constraint Logic Programming With Negation (full version)., CoRR abs/2507.16067 (2025). doi:10.48550/ARXIV.2507.16067.

[2] J. Spaans, J. Heyninck, A Unifying Framework for Semiring-Based Constraint Logic Programming With Negation., in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025, 2025, pp. 2684–2692. doi:10.24963/IJCAI.2025/299.

[3] S. Bistarelli, Semirings for Soft Constraint Solving and Programming, volume 2962 of *Lecture Notes in Computer Science*, Springer, 2004. doi:10.1007/B95712.

[4] S. Bistarelli, U. Montanari, F. Rossi, Semiring-based constraint satisfaction and optimization., J. ACM 44 (1997) 201–236. doi:10.1145/256303.256306.

[5] S. Bistarelli, U. Montanari, F. Rossi, Semiring-based contstraint logic programming: Syntax and semantics., ACM Trans. Program. Lang. Syst. 23 (2001) 1–29. doi:10.1145/383721.383725.

[6] V. Belle, L. D. Raedt, Semiring programming: A semantic framework for generalized sum product problems., Int. J. Approx. Reason. 126 (2020) 181–201. doi:10.1016/J.IJAR.2020.08.001.

[7] T. Eiter, R. Kiesel, ASP(AC): Answer Set Programming with Algebraic Constraints., Theory Pract. Log. Program. 20 (2020) 895–910. doi:10.1017/S1471068420000393.

[8] T. Eiter, R. Kiesel, Weighted LARS for Quantitative Stream Reasoning., in: 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), 2020, pp. 729–736. doi:10.3233/FAIA200160.

[9] T. J. Green, G. Karvounarakis, V. Tannen, Provenance semirings., in: Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China, 2007, pp. 31–40. doi:10.1145/1265530.1265535.

[10] M. A. Khamis, H. Q. Ngo, R. Pichler, D. Suciu, Y. R. Wang, Convergence of datalog over (Pre-) Semirings., J. ACM 71 (2024) 8:1–8:55. doi:10.1145/3643027.

[11] A. Kimmig, G. V. den Broeck, L. D. Raedt, An Algebraic Prolog for Reasoning about Possible Worlds., in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011, 2011, pp. 209–214. doi:10.1609/AAAI.V25I1.7852.

[12] A. Kimmig, G. V. den Broeck, L. D. Raedt, Algebraic model counting., J. Appl. Log. 22 (2017) 46–62. doi:10.1016/J.JAL.2016.11.031.

[13] K. Gödel, Zum intuitionistischen aussagenkalkül, Anzeiger der Akademie der Wissenschaften in Wien 69 (1932) 65–66.

[14] M. Denecker, V. Marek, M. Truszczynski, Approximations, Stable Operators, Well-Founded Fixpoints And Applications In Nonmonotonic Reasoning, Logic-based Artificial Intelligence (2001). doi:10.1007/978-1-4615-1567-8_6.
[15] M. A. Khamis, H. Q. Ngo, R. Pichler, D. Suciu, Y. R. Wang, Convergence of Datalog over (Pre-) Semirings., SIGMOD Rec. 52 (2023) 75–82. doi:10.1145/3604437.3604454.
[16] J. Vennekens, D. Gilis, M. Denecker, Splitting an operator: Algebraic modularity results for logics with fixpoint semantics, CoRR cs.AI/0405002 (2004).
[17] J. Heyninck, An Algebraic Notion of Conditional Independence, and Its Application to Knowledge Representation (full version)., CoRR abs/2412.13712 (2024). doi:10.48550/ARXIV.2412.13712.
[18] J. Heyninck, O. Arieli, B. Bogaerts, Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming., Artif. Intell. 331 (2024) 104110. doi:10.1016/J.ARTINT.2024.104110.