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

Jeroen Spaans[1] and **Jesse Heyninck**[2]
[1]Open Universiteit, The Netherlands
[1,2]University of Cape Town, South Africa
{jeroen.spaans, jesse.heyninck}@ou.nl

Authors' Note

In this document you will find an extended abstract of a full paper accepted at IJCAI 2025. A complete copy of the work, including proofs and appendices, will be made freely available on arXiv prior to the event.

1 Extended Abstract

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.

```
1  % We define two tasks,
2  % taking 6 and 4 hours to complete respectively.
3  task(t1, 6).
4  task(t2, 4).
5  % A task is completed if enough time is scheduled.
6  completed(Task, HoursScheduled) :-
7     task(Task, HoursRequired),
8     HoursRequired ≤ HoursScheduled.
9  % We set a time limit of eight hours.
10 inTimeLimit(Hours1, Hours2) :-
11    Hours1 + Hours2 ≤ 8.
12 % A schedule is evaluated by the degree to which both tasks
13 % are completed and the total allotted time does not exceed
14 % the time available.
15 schedule(HoursTask1, HoursTask2) :-
16    completed(t1, HoursTask1),
17    completed(t2, HoursTask2),
18    inTimeLimit(HoursTask1, HoursTask2).
```

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 ≤ 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. [Bistarelli, 2004; Bistarelli *et al.*, 1997; Bistarelli *et al.*, 2001] 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 [Belle and Raedt, 2020; Eiter and Kiesel, 2020a; Eiter and Kiesel, 2020b; Green *et al.*, 2007; Khamis *et al.*, 2024; Kimmig *et al.*, 2011; Kimmig *et al.*, 2017] 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 [Gödel, 1932], and also used in the semiring-based formalism of Eiter and Kiesel [Eiter and Kiesel, 2020b]—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 [Denecker *et al.*, 2001], 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: first, 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. The minimal model semantics based on these notions of model are then compared with the least fixpoint semantics based on an immediate consequence operator. Next, we investigate 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—and find it to be a generalization of the method studied by Bistarelli et al. 2001. Then, we study a generalized notion of negation appearing at various points in the literature as applied to our semiring-based framework. Finally, we apply approximation fixpoint theory to our immediate consequence operator—made nonmonotonic 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 [Bistarelli *et al.*, 2001] and [Khamis *et al.*, 2023]. Computational complexity, implementations, and applying AFT-based notions such as stratification [Vennekens *et al.*, 2004], conditional independence [Heyninck, 2024] and non-determinism [Heyninck *et al.*, 2024] are left as future work.

References

[Belle and Raedt, 2020] Vaishak Belle and Luc De Raedt. Semiring programming: A semantic framework for generalized sum product problems. *Int. J. Approx. Reason.*, 126:181–201, 2020.

[Bistarelli *et al.*, 1997] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint satisfaction and optimization. *J. ACM*, 44(2):201–236, 1997.

[Bistarelli *et al.*, 2001] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based contstraint logic programming: Syntax and semantics. *ACM Trans. Program. Lang. Syst.*, 23(1):1–29, 2001.

[Bistarelli, 2004] Stefano Bistarelli. *Semirings for Soft Constraint Solving and Programming*, volume 2962 of *Lecture Notes in Computer Science*. Springer, 2004.

[Denecker *et al.*, 2001] Marc Denecker, Victor Marek, and Miros Truszczynski. Approximations, Stable Operators, Well-Founded Fixpoints And Applications In Nonmonotonic Reasoning. *Logic-based Artificial Intelligence*, December 2001.

[Eiter and Kiesel, 2020a] Thomas Eiter and Rafael Kiesel. ASP(AC): Answer Set Programming with Algebraic Constraints. *Theory Pract. Log. Program.*, 20(6):895–910, 2020.

[Eiter and Kiesel, 2020b] Thomas Eiter and Rafael 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)*, pages 729–736, 2020.

[Green *et al.*, 2007] Todd J. Green, Gregory Karvounarakis, and Val 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*, pages 31–40, 2007.

[Gödel, 1932] Kurt Gödel. Zum intuitionistischen aussagenkalkul. *Anzeiger Akademie der Wissenschaften Wien, mathematisch-naturwissenschaftliche Klasse*, 69:65–66, 1932.

[Heyninck *et al.*, 2024] Jesse Heyninck, Ofer Arieli, and Bart Bogaerts. Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming. *Artificial Intelligence*, 331:104110, 2024.

[Heyninck, 2024] Jesse Heyninck. An algebraic notion of conditional independence, and its application to knowledge representation (full version). *arXiv preprint arXiv:2412.13712*, 2024.

[Khamis *et al.*, 2023] Mahmoud Abo Khamis, Hung Q. Ngo, Reinhard Pichler, Dan Suciu, and Yisu Remy Wang. Convergence of Datalog over (Pre-) Semirings. *SIGMOD Rec.*, 52(1):75–82, 2023.

[Khamis *et al.*, 2024] Mahmoud Abo Khamis, Hung Q. Ngo, Reinhard Pichler, Dan Suciu, and Yisu Remy Wang. Convergence of datalog over (Pre-) Semirings. *J. ACM*, 71(2):8:1–8:55, April 2024.

[Kimmig *et al.*, 2011] Angelika Kimmig, Guy Van den Broeck, and Luc De 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*, pages 209–214, 2011.

[Kimmig *et al.*, 2017] Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. Algebraic model counting. *J. Appl. Log.*, 22:46–62, 2017.

[Vennekens *et al.*, 2004] Joost Vennekens, David Gilis, and Marc Denecker. Splitting an operator: Algebraic modularity results for logics with fixpoint semantics. *CoRR*, cs.AI/0405002, 2004.