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

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

Abstract

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, 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.

1 Introduction

Constraint logic programming (CLP) facilitates solving problems requiring the consideration of constraints, and is well-established as a fruitful formalism over the last decades. To further increase its expressivity, its language has been extended to take into account information about a problem domain of a more quantitative nature, e.g. by allowing fuzzy constraint satisfaction or taking into account scores or penalties. Due to the wide variety of such possible extensions, several works have sought to unify these approaches using the algebraic concept of a *semiring* [Bistarelli *et al.*, 2001; Khamis *et al.*, 2023]. Even though these works represent a big step forward in terms of unification, they introduce their own assumptions on semirings, which makes it hard to compare these unifying approaches. Furthermore, they restrict attention to positive logic programs, i.e. programs consisting of rules without negation in the body. This is unfortunate, as it is well-known from e.g. answer set programming that negation increases both the conceptual and computational expressivity. However, it also introduces non-monotonicity of the corresponding immediate consequence operator, leading to intricacies in the definition of semantics.

Approximation fixpoint theory (AFT) [Denecker *et al.*, 2001] is a unifying framework for the definition and study of semantics of non-monotonic formalisms. It is a purely algebraic theory which was shown to unify the semantics of, among others, logic programming, default logic, and autoepistemic logic. Due to the algebraic constructions defined in AFT, it often suffices to define a so-called approximating operator to derive a whole class of well-behaved semantics, including the so-called stable and well-founded semantics.

In this paper, we apply AFT to unify and generalize existing approaches to semiring-based constraint logic programming. In more detail, the contributions of this paper are the following: (1) we unify the approaches of [Bistarelli *et al.*, 2001; Khamis *et al.*, 2023] in the framework of AFT, making clear the different assumptions used in these approaches and how they affect the behaviour of the corresponding semantics, (2) we generalize the language of semiring-based constraint logic programming to allow for negation in the body of rules, and (3) use AFT to define and study semantics for such programs, introducing among others the stable and well-founded semantics, and show how they generalize both existing approaches to (positive) SCLPs and normal logic programs (nlps).

Outline of the Paper: In Section 2 we provide background on lattice theory, AFT and semirings. Section 3 considers properties of semirings and derived orders. Section 4 reconstructs existing semantics for SCLPs with lifted assumptions, and in Section 5, we introduce the syntax and semantics of negation, whereas in Section 6 the semantics for programs with negation are studied. We conclude with regard to related work in Section 7.

2 General Background

In this section, we introduce the necessary preliminaries on lattice theory (Section 2.1), approximation fixpoint theory (Section 2.2) and semirings (Section 2.3).

2.1 Lattice Theory

A *partially ordered set (poset)* L is an ordered pair $\langle \mathcal{L}, \leq \rangle$, where $\leq \subseteq \mathcal{L} \times \mathcal{L}$ is a reflexive, antisymmetric and transitive relation. Given a poset $L = \langle \mathcal{L}, \leq \rangle$ with $S \subseteq \mathcal{L}$, A *lower bound* (resp. *upper bound*) of S is an element $x \in \mathcal{L}$ such that $x \leq s$ (resp. $s \leq x$) for all $s \in S$. A lower bound y of S is the unique *greatest lower bound* (glb) of S, denoted $\bigwedge S$, if for every lower bound x of S in $\mathcal{L}$ we have that $x \leq y$,

and similarly for the *least upper bound* (lub). We also denote $x \wedge y$ by $\bigwedge\{x, y\}$ and $x \vee y$ by $\bigvee\{x, y\}$.

A *lattice* is a partially ordered set such that each pair of elements has a greatest lower bound and a least upper bound. A lattice is *bounded* if there are $\bot, \top \in \mathcal{L}$ such that for any $x \in \mathcal{L}$, $\bot \leq x \leq \top$ holds, and it is complete if every subset of $\mathcal{L}$ has a greatest lower bound and a least upper bound. $\bot_{\leq}$ denotes a complete lattice's least element, and $\top_{\leq}$ its greatest element. While every complete lattice is a bounded lattice, it should be noted that the converse is not true [Blyth, 2005].

We will study operators $\mathrm{O} : \mathcal{L} \to \mathcal{L}$, which are *$\leq$-monotone* if for all $x \leq y$ we have that $\mathrm{O}(x) \leq \mathrm{O}(y)$, and *$\leq$-antimonotone* if for all $x \leq y$ we have that $\mathrm{O}(y) \leq \mathrm{O}(x)$.[1] This work also concerns commutative binary operators $\mathrm{O} : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$. Such operators are called (anti)monotone if they are monotone (resp. antimonotone) in either operand. That is, we fix one operand at a time and consider the (anti)monotonicity of $x \mapsto \mathrm{O}(x, y)$ and $y \mapsto \mathrm{O}(x, y)$ separately. An element $x \in \mathcal{L}$ is a *prefixpoint* of O if $\mathrm{O}(x) \leq x$, a *fixpoint* of O if $\mathrm{O}(x) = x$, and a *postfixpoint* of O if $x \leq \mathrm{O}(x)$. Should it exist, $\mathrm{lfp}_{\leq}(\mathrm{O})$ denotes the *$\leq$-least fixpoint* of O.[2]

Thanks to Theorem 5.1 of Cousot and Cousot [Cousot and Cousot, 1979], we have a constructive characterization of this least fixpoint [Tarski, 1955]. For this characterization, we first define the ordinal powers of a function $\mathrm{O} : \mathcal{L} \to \mathcal{L}$ by defining $\mathrm{O}^0(x) = x$, $\mathrm{O}^{\alpha+1}(x) = \mathrm{O}(\mathrm{O}^{\alpha}(x))$ for a successor ordinal α, and $\mathrm{O}^{\alpha}(x) = \bigvee_{\beta<\alpha} \mathrm{O}^{\beta}(x)$ for a limit ordinal α. The least fixpoint of a monotone operator on a complete lattice can be constructed by applying its ordinal powers starting from $\bot_{\leq}$:

Theorem 1 ([Cousot and Cousot, 1979])**.** *Given a complete lattice $L = \langle \mathcal{L}, \leq \rangle$ and a $\leq$-monotone operator $\mathrm{O} : \mathcal{L} \to \mathcal{L}$, there is an ordinal α such that $\mathrm{O}^{\alpha}(\bot_{\leq})$ is the least fixpoint of* O*, which coincides with* O*'s least prefixpoint.*

2.2 Approximation Fixpoint Theory

In this section, we recall *approximation fixpoint theory* (AFT) [Denecker *et al.*, 2001], which offers techniques for approximating the fixpoints of a possibly non-monotonic operator.

Given a lattice $L = \langle \mathcal{L}, \leq \rangle$, a *bilattice* is the structure $L^2 = \langle \mathcal{L}^2, \leq_{\mathrm{p}}, \leq_{\mathrm{v}} \rangle$, where $\mathcal{L}^2 = \mathcal{L} \times \mathcal{L}$, and for every $x_1, y_1, x_2, y_2 \in \mathcal{L}$, $(x_1, y_1) \leq_{\mathrm{p}} (x_2, y_2)$ if $x_1 \leq x_2$ and $y_2 \leq y_1$, and $(x_1, y_1) \leq_{\mathrm{v}} (x_2, y_2)$ if $x_1 \leq x_2$ and $y_1 \leq y_2$. Intuitively, bilattice elements $(x, y) \in \mathcal{L}^2$ approximate elements in the interval $[x, y] = \{z \in \mathcal{L} : x \leq z \leq y\}$. If $x \leq y$, we call (x, y) *consistent*, and $\mathcal{L}^c$ is the set of all consistent pairs. We call elements $(x, x) \in \mathcal{L}^c$ *exact*, and note that the set of all exact elements constitutes an embedding of $\mathcal{L}$ in $\mathcal{L}^2$. We also use projection functions $(x, y)_1 = x$ and $(x, y)_2 = y$.

An operator $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ is an *approximator of* $\mathrm{O} : \mathcal{L} \to \mathcal{L}$ if it is $\leq_{\mathrm{p}}$-monotone and, for any $x \in \mathcal{L}$, $\mathcal{O}(x, x) = (\mathrm{O}(x), \mathrm{O}(x))$. Approximating operators $\mathcal{O}$ can be thought of as combinations of two separate operators $(\mathcal{O}(\cdot, \cdot))_1$ and $(\mathcal{O}(\cdot, \cdot))_2$ calculating, respectively, a lower and upper bound for the value of the approximated operator O. We denote $(\mathcal{O}(\cdot, \cdot))_1$ and $(\mathcal{O}(\cdot, \cdot))_2$ as $\mathcal{O}_{\mathrm{l}}(x, y)$ and $\mathcal{O}_{\mathrm{u}}(x, y)$ respectively. Specifically, we have $\mathcal{O}_{\mathrm{l}}(\cdot, y) = \lambda x.\, \mathcal{O}_{\mathrm{l}}(x, y)$ (i.e., $\mathcal{O}_{\mathrm{l}}(\cdot, y)(x) = \mathcal{O}_{\mathrm{l}}(x, y)$) and $\mathcal{O}_{\mathrm{u}}(x, \cdot) = \lambda y.\, \mathcal{O}_{\mathrm{u}}(x, y)$.

Given a complete lattice $\langle \mathcal{L}, \leq \rangle$ and an approximating operator $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$, the *stable operator* for $\mathcal{O}$ is $\mathcal{S}(\mathcal{O})(x, y) = (\mathrm{lfp}(\mathcal{O}_{\mathrm{l}}(\cdot, y)), \mathrm{lfp}(\mathcal{O}_{\mathrm{u}}(x, \cdot)))$. For any $\leq_{\mathrm{p}}$-monotone operator $\mathcal{O}$ on $\mathcal{L}^2$, the fixpoints of $\mathcal{S}(\mathcal{O})$ are $\leq_{\mathrm{v}}$-minimal fixpoints of $\mathcal{O}$ [Denecker *et al.*, 2001]. Also, $\mathcal{O}_{\mathrm{l}}(\cdot, y)$ and $\mathcal{O}_{\mathrm{u}}(x, \cdot)$ are $\leq$-monotone operators —guaranteeing $\mathcal{S}(\mathcal{O})$ is well-defined, and $\mathcal{S}(\mathcal{O})$ is $\leq_{\mathrm{p}}$-monotone [Denecker *et al.*, 2001].

Given an approximating operator $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$. (x, y) is: the *Kripke-Kleene (KK) fixpoint* of $\mathcal{O}$ if $(x, y) = \mathrm{lfp}_{\leq_{\mathrm{p}}}(\mathcal{O})$; a *stable* fixpoint of $\mathcal{O}$ if $(x, y) = \mathcal{S}(\mathcal{O})(x, y)$; the *well-founded (WF) fixpoint* of $\mathcal{O}$ if $(x, y) = \mathrm{lfp}_{\leq_{\mathrm{p}}}(\mathcal{S}(\mathcal{O}))$.

A lattice operator may permit multiple approximating operators, but Denecker et al. 2004 define the most precise approximating operator $\mathcal{O}^{\mathrm{u}}$ approximating an operator O, called the ultimate approximator of O. In more detail, given a complete lattice $L = \langle \mathcal{L}, \leq \rangle$, and $\mathrm{O} : \mathcal{L} \to \mathcal{L}$, the *ultimate approximator* of O, $\mathcal{O}^{\mathrm{u}} : \mathcal{L}^c \to \mathcal{L}^c$ is defined by $\mathcal{O}^{\mathrm{u}}(x, y) = (\bigwedge \mathrm{O}([x, y]), \bigvee \mathrm{O}([x, y]))$.

2.3 Semirings

In this section we recall the necessary preliminaries on semirings, which we will use as the range of interpretations of logic programs. A *monoid* is a tuple $M = \langle \mathcal{L}, \circ, e \rangle$ such that $\circ : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$ is a binary, associative operator on $\mathcal{L}$; and we have for any $x \in \mathcal{L}$ that $e \circ x = x = x \circ e$—$e$ is the identity element of $\circ$. If $\circ$ is commutative, we call M a *commutative monoid*. If $\langle \mathcal{L}, \leq \rangle$ is a poset s.t. $\circ$ is $\leq$-monotone, M is *ordered* by $\leq$, and if for every $x \in \mathcal{L}$ we have that $e \leq x$, it is *positively ordered*.

A commutative monoid $M = \langle \mathcal{L}, \circ, e \rangle$ is equipped with a 'sum' operation $\bigcirc_I$ for finite index sets I such that: $\bigcirc_{i \in \emptyset} x_i = e$; $\bigcirc_{i \in \{j\}} x_i = x_j$; $\bigcirc_{i \in \{j,k\}} x_i = x_j \circ x_k$ for $j \neq k$; and $\bigcirc_{j \in J} \bigcirc_{i \in I_j} x_i = \bigcirc_{i \in I} x_i$ if $\bigcup_{j \in J} I_j = I$ and $I_j \cap I_{j'} = \emptyset$ for $j \neq j'$. If the sum operation $\bigcirc_I$ is well-defined for any infinite index set I, M is a *complete monoid*. [3]

A *semiring* is a tuple $S = \langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$, such that: $\langle \mathcal{L}, +, \mathbf{0} \rangle$ is a commutative monoid, with $+$ being called the additive operator; $\langle \mathcal{L}, \times, \mathbf{1} \rangle$ is a monoid, with $\times$ being called the multiplicative operator; we have for any $x \in \mathcal{L}$ that $x \times \mathbf{0} = \mathbf{0} = \mathbf{0} \times x$—i.e. $\mathbf{0}$ is the absorbing element of $\times$; and we have for any $x, y, z \in \mathcal{L}$ that $x \times (y + z) = (x \times y) + (x \times z)$ and $(y + z) \times x = (y \times x) + (z \times x)$—i.e. $\times$ left- and right-distributes over $+$. *Sum* ($\sum$) and *product* ($\prod$) denote the sum operation of $+$ and $\times$ respectively. We identify six specific classes of semirings:

1. If $\langle \mathcal{L}, \times, \mathbf{1} \rangle$ is a commutative monoid, S is a *commutative semiring*.
2. If $x + x = x$ holds for any $x \in \mathcal{L}$, S is an *idempotent semiring*.
3. If S is a commutative idempotent semiring, and we have for any $x \in \mathcal{L}$ that $x + \mathbf{1} = \mathbf{1} = \mathbf{1} + x$—i.e. $\mathbf{1}$ is the

[1] When the ordering clear, we simply say O is *(anti)monotone*.
[2] If the ordering is clear, we simply write $\mathrm{lfp}(\mathrm{O})$.
[3] Bijective indexings are left implicit. E.g. $\bigcirc\{x_1, \ldots, x_n\} = x_1 \circ \ldots \circ x_n$ and $\bigcirc(x_1, x_2, \ldots) = x_1 \circ x_2 \circ \ldots$ (if it is defined).

absorbing element of $+$—then S is a *constraint-based semiring (c-semiring)* [Bistarelli *et al.*, 2001].

4. S is a *complete semiring* if $\langle \mathcal{L}, +, \mathbf{0} \rangle$ is a complete monoid, and it holds that $\sum_{i \in I}(x \times x_i) = x \times (\sum_{i \in I} x_i)$ and $\sum_{i \in I}(x_i \times x) = (\sum_{i \in I} x_i) \times x$.
5. If $\langle \mathcal{L}, +, \mathbf{0} \rangle$ and $\langle \mathcal{L}, \times, \mathbf{1} \rangle$ are ordered by poset $\langle \mathcal{L}, \leq \rangle$, S is *ordered* by $\langle \mathcal{L}, \leq \rangle$.
6. If $\langle \mathcal{L}, +, \mathbf{0} \rangle$ is positively ordered by the poset $\langle \mathcal{L}, \leq \rangle$ and $\langle \mathcal{L}, \times, \mathbf{1} \rangle$ is ordered by $\langle \mathcal{L}, \leq \rangle$, then S is *positively ordered* by $\langle \mathcal{L}, \leq \rangle$.

Note. Hereafter, unless otherwise specified, $S = \langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is assumed to be a commutative semiring.

Some examples of commutative semirings are:

- $\mathbb{S} = \langle \mathbb{F}, +, \cdot, 0, 1 \rangle$, for $\mathbb{F} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$;
- $\mathcal{P}(A) = \langle 2^A, \bigcup, \bigcap, \emptyset, A \rangle$, the power set semiring ;
- $\mathbb{B} = \langle \{\textit{true}, \textit{false}\}, \vee, \wedge, \textit{false}, \textit{true} \rangle$, the Boolean semiring;
- $\mathbb{F} = \langle [0, 1], \max, \min, 0, 1 \rangle$, the fuzzy semiring;
- $\mathbb{N}_\infty = \langle \mathbb{N} \bigcup \{\infty\}, +, \cdot, 0, 1 \rangle$, where $n + \infty = \infty$ for all n and $m \cdot \infty = \infty$ for all $m \neq 0$ — the semiring of natural numbers lifted to infinity.

3 Families of Semirings and Their Orderings

In Section 2.1, we saw a constructive definition for the least fixpoint of a monotone operator on a complete lattice, and, in Section 2.2, we saw that such constructions are widely used in AFT. In the following sections, we will use such constructions again to define the semantics of our semiring-based formalism. For this, we will require complete lattices that order our semirings. Some semirings of interest have standard orderings that happen to be complete lattices—for example, the Boolean semiring $\mathbb{B}$ forms a complete lattice under *false* $<$ *true* and the power set semiring $\mathcal{P}(A)$ forms a complete lattice under set inclusion. Such orderings are not readily apparent for all semirings, however. We therefore investigate a "semiring-agnostic" ordering and look for conditions under which this ordering forms a complete lattice. This ordering, found throughout the literature [Green *et al.*, 2007; Khamis *et al.*, 2024; Hannula, 2023], is the natural order.

Definition 1. For any $x, y \in \mathcal{L}$, we say $x \leq_{\mathrm{N}} y$ if there exists a $z \in \mathcal{L}$ such that $x + z = y$. When $\leq_{\mathrm{N}}$ is a partial order, we call it the *natural order*. A semiring $\langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$ for which $\langle \mathcal{L}, \leq_{\mathrm{N}} \rangle$ is a partial order is called *naturally ordered*.

Not all semirings are naturally ordered. When semirings allow for additive inverses, the ordering is not antisymmetric. In all other cases, the ordering is a partial order.

Lemma 1. *S is naturally ordered if and only if it contains* no *elements x and $-x$ s.t. $x \neq \mathbf{0}$ and $x + (-x) = \mathbf{0}$.*

Our additive and multiplicative operators are monotone for naturally ordered semirings.

Lemma 2. *If S is naturally ordered, it is ordered by $\leq_{\mathrm{N}}$.*

A complete lattice of semiring elements must be bounded. For naturally ordered semirings, the element $\mathbf{0}$ is the least element bounding our lattice from below.

Lemma 3. *If S is naturally ordered, $\bot_{\leq_{\mathrm{N}}} = \mathbf{0}$.*

While all naturally ordered semirings are bounded from below by $\mathbf{0}$, they are not generally bounded from above. A straightforward example of this is the whole number semiring $\mathbb{N}$, for which the natural order coincides with the usual ordering and which extends infinitely upwards. Naturally, this also means not all naturally ordered semirings form a complete lattice. In fact, a naturally ordered semiring has a maximal element if and only if said element is absorbing for $+$.

Lemma 4. *If S is naturally ordered, $\top_{\leq_{\mathrm{N}}} = x \in \mathcal{L}$ if and only if x is absorbing for $+$.*

Bistarelli et al. 2001 propose an ordering specific to c-semirings which forms a complete lattice under certain circumstances.

Definition 2. Let S be a c-semiring. For any $a, b \in \mathcal{L}$, we define the *c-semiring order* as $a \leq_{\mathrm{C}} b$ if $a + b = b$.

We treat $\leq_{\mathrm{C}}$ as if it were defined for any (non-c-)semiring and find it to be a specific case of the natural order.

Lemma 5. *For idempotent S and $a, b \in \mathcal{L}$: $a \leq_{\mathrm{N}} b$ iff $a \leq_{\mathrm{C}} b$.*

Not only do $\leq_{\mathrm{N}}$ and $\leq_{\mathrm{C}}$ coincide whenever S is idempotent—in this case they are also always a partial order. This, of course, also means that idempotent semirings do not permit additive inverses.

Lemma 6. *If S is idempotent, it is naturally ordered.*

Bistarelli et al. [1997] show how $\leq_{\mathrm{N}}$ can form a complete lattice for c-semirings by first showing that $+$ and $\vee$ coincide, then showing that any set that has the $\vee$ also has the $\wedge$, and finally combining these two facts to get that any set has both the $\vee$ and $\wedge$. However, this reasoning relies on the assumption that the sum is defined for any (infinite) set of c-semiring elements. This assumption does not hold for every c-semiring, as shown in Appendix A.1. We also find that a complete c-semiring is a stronger assumption than necessary to obtain these results. Indeed, we find the same results for complete idempotent semirings with a greatest element $\top_{\leq_{\mathrm{N}}}$.

Lemma 7. *If S is a complete idempotent semiring s.t. $\top_{\leq_{\mathrm{N}}}$ exists, then $\langle \mathcal{L}, \leq_{\mathrm{N}} \rangle$ is a complete lattice.*

Completeness of a semiring, by itself, is not sufficient to guarantee a complete lattice of semiring elements.

Lemma 8. *Not every complete semiring gives rise to a complete lattice under the natural order.*

Other examples of a semiring inducing a complete lattice under the natural order are $\mathbb{N}_\infty$ and $\langle [0, 1], +, \cdot, 0, 1 \rangle$.

4 Semiring-based Constraint Logic Programming

We begin our study of semiring-based constraint logic programming, limiting ourselves to the positive fragment of the full formalism that will be presented in Section 5. In essence, this special case is a simplified account of semiring-based constraint logic programs as introduced by Bistarelli et al. 2001, in which we exclude variables and functions from our language, leaving such extensions for future work. On the other hand, we lift their assumption of c-semirings and consider general semirings.

Definition 3. Given a semiring $S = \langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$ and a set of atoms $\mathcal{A}$, *generalized atoms* are defined as atoms and semiring values. A *clause* is an expression of the form H :- $B_1, \ldots, B_n$ where H, the *head* of the clause, is an atom and each B_i appearing in the (possibly empty) *body* of the clause is a generalized atom. A *positive semiring-based constraint logic program* for S (*PSCLP(S)*, for short[4]), is a set of clauses. $\mathcal{A}_P$ denotes the atoms used in PSCLP(S) P.

Notice that Datalog programs [Ceri *et al.*, 1989] are a special case of PSCLPs, namely the programs PSCLP($\mathbb{B}$).

Example 1. We consider the following program P (adapted from [Bistarelli *et al.*, 2001]) as a running example.

```
c1: solution(a) :- path(a,b).
c2: solution(a) :- path(a,c).
c3: path(a,b) :- mass_transit(a).
c4: path(a,c) :- car(a).
c5: mass_transit(a) :- train(a).
c6: train(a) :- 2.
c7: car(a) :- 3.
```

The semiring over which P is defined, from which the integer values in the program stem, is the optimization semiring $\mathbb{O} = \langle \mathbb{N} \bigcup \{\infty\}, \min, +, \infty, 0 \rangle$. This semiring allows us to optimize over constraints, by assigning each constraint an integer representing the cost of the connection it models, combining constraints by summing their costs, and comparing constraints using the minimum operator. We order the semiring with the complete lattice $\langle \mathbb{N} \bigcup \{\infty\}, \geq \rangle$—higher costs are 'lesser than' lower costs.

The clauses c_6 and c_7 describe the costs associated with travelling by train or car respectively. c_5 tells us that train travel is a form of mass transit. Since `mass_transit(a)` is only defined by c_5, the cost of using mass transit will only depend on the cost of travelling by train. The clauses c_3 and c_4 inform us that mass transit and car travel each allow us to take a different path, and that the costs associated with following these paths will depend on the cost of using their corresponding mode of transport. Finally, the clauses c_1 and c_2 tell us that either path constitutes a solution to our problem, and that we must compare the costs of either path to obtain the optimal solution to our problem.

The semantics of PSCLPs is given in terms of interpretations that assign semiring values to formulas, essentially generalizing classical interpretations by allowing any semiring value as truth value instead of just the Boolean ones.

Definition 4. An *interpretation* I of some PSCLP P based on a semiring $S = \langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is a mapping $\mathrm{I} : \mathcal{A}_P \to \mathcal{L}$. We extend interpretations to semiring element $x \in \mathcal{L}$ by $\mathrm{I}(x) = x$, to conjunctions of formulas (inductively) by $\mathrm{I}(A, B) = \mathrm{I}(A) \times \mathrm{I}(B)$, and to empty conjunctions by $\mathrm{I}(\emptyset) = \mathbf{1}$.

Note. Hereafter, unless otherwise specified, we assume that $\langle \mathcal{L}, \leq \rangle$ is a partial order, and that P is a PSCLP(S).

We endow the set of P's interpretations IS_P with an ordering that respects the partial order of semiring elements.

[4]We write PSCLP when the semiring is clear or unimportant.

Definition 5. The *partial order of interpretations* $\langle IS_P, \preceq \rangle$ is derived from $\langle \mathcal{L}, \leq \rangle$ such that for any $\mathrm{I}_1, \mathrm{I}_2 \in IS_P$, $\mathrm{I}_1 \preceq \mathrm{I}_2$ if $\mathrm{I}_1(x) \leq \mathrm{I}_2(x)$ for any atom $x \in \mathcal{A}_P$.

When $\langle \mathcal{L}, \leq \rangle$ is a complete lattice, so is $\langle IS_P, \preceq \rangle$. We now introduce the notion of model for PSCLPs.

Definition 6. $\mathrm{I} \in IS_P$ is a *semiring model* of P if it holds for every $H \in \mathcal{A}_P$ that $\sum\{\prod_{i=1}^{n} \mathrm{I}(B_i) \mid H$:- $B_1, \ldots, B_n \in P\} \leq \mathrm{I}(H)$. MS_P denotes the set of semiring models of P.

This notion of model deviates slightly from the traditional notion of model. The traditional notion of model requires only that the head of each clause be interpreted as no lesser than the clause's body—considering every clause independently. Our notion of model instead considers all clauses defining an atom in unison, requiring that said atom be interpreted as no lesser than the sum of these clauses' bodies. The difference between these notions stems from the fact that the operators used traditionally for the comparison between clauses (like $\vee$ or $\max$) are idempotent, while the additive operators used for this purpose in our framework are not necessarily. In this way, our notion of model treats semirings more seriously as a structure for evaluation.[5]

Example 2. We illustrate which interpretations are semiring models of our running example program. For interpretation $\mathrm{I} \in IS_P$ to be a semiring model of P, we require that:

- $3 \geq \mathrm{I}(\mathtt{car(a)})$ and $2 \geq \mathrm{I}(\mathtt{train(a)})$;
- $\mathrm{I}(\mathtt{train(a)}) \geq \mathrm{I}(\mathtt{mass\_transit(a)})$;
- $\mathrm{I}(\mathtt{car(a)}) \geq \mathrm{I}(\mathtt{path(a,c)})$;
- $\mathrm{I}(\mathtt{mass\_transit(a)}) \geq \mathrm{I}(\mathtt{path(a,b)})$; and
- $\min\{\mathrm{I}(\mathtt{path(a,c)}), \mathrm{I}(\mathtt{path(a,b)})\} \geq \mathrm{I}(\mathtt{solution(a)})$.

Given all the interpretations of a PSCLP, we would like to select a single unique model as the representative one. In logic programming, this representative model is the minimal model [Lloyd, 1987], obtained as the intersection of each model's sets of true atoms. Intuitively, this model minimizes truth ascription while still satisfying all clauses. In other words, it makes true what must be true and nothing else. Here, we follow a similar approach but require a generalization of the notion of model intersection.

Definition 7. Given a complete lattice $\langle \mathcal{L}, \leq \rangle$ and $A \in \mathcal{A}_P$, we define the *model intersection* of $M \subseteq MS_P$ as $\bigcap M(A) = \bigwedge\{\mathrm{I}(A) \mid \mathrm{I} \in M\}$.

Lemma 9. *If S is ordered by a complete lattice $\langle \mathcal{L}, \leq \rangle$ and $M \subseteq MS_P$, $\bigcap M$ is a semiring model for PSCLP(S) P.*

We obtain the minimal semiring model by intersecting MS_P.

Definition 8. The *minimal semiring model* of PSCLP P is $M_P^s = \bigcap MS_P$. M_P^s is the model-theoretic semantics of P.

Example 3. For our running example, the minimal semiring model gives us the following.

- $M_P^s(\mathtt{car(a)}) = 3$ and $M_P^s(\mathtt{train(a)}) = 2$;
- $M_P^s(\mathtt{mass\_transit(a)}) = 2$;

[5]We defer an investigation of the differences between these notions of model to Appendix A.2.

	I_1	I_2	I_3	I_4
`train(a)`	2	2	2	2
`car(a)`	3	3	3	3
`mass_transit(a)`	∞	2	2	2
`path(a,c)`	∞	3	3	3
`path(a,b)`	∞	∞	2	2
`solution(a)`	∞	∞	3	2

Table 1: $\mathrm{T_P}$ applied to Ex. 4, where $\mathrm{I}_1 = \mathrm{T_P}(\bot_\preceq)$ and $\mathrm{I}_{i+1} = \mathrm{T_P}(\mathrm{I}_i)$ for $i > 0$.

- $M_P^s(\mathtt{path(a,c)}) = 3$;
- $M_P^s(\mathtt{path(a,b)}) = 2$; and
- $M_P^s(\mathtt{solution(a)}) = 2$.

The model-theoretic semantics of a PSCLP lacks a constructive definition. We therefore work toward an equivalent fixpoint semantics that does have a constructive definition, based on an extension of the well-known immediate consequence operator from Datalog to PSCLPs.

Definition 9. Given interpretation $\mathrm{I} \in IS_P$ and atom $H \in \mathcal{A}_P$, $\mathrm{T_P}(\mathrm{I})(H) = \sum\{\prod_{i=1}^n \mathrm{I}(B_i) | H \text{ :- } B_1, \ldots, B_n \in P\}$.

The fixpoint semantics of a PSCLP is defined as the least fixpoint of this immediate consequence operator. This fixpoint can be constructed if $\mathrm{T_P}$ is monotone on a complete lattice.

Lemma 10. *If S is ordered by $\langle \mathcal{L}, \leq \rangle$, $\mathrm{T_P}$ is $\preceq$-monotone.*

Lemma 7 now gives us sufficient conditions for the existence of a constructive definition of our fixpoint semantics.

Corollary 1. *If S is ordered by the complete lattice $\langle \mathcal{L}, \leq \rangle$ (e.g. when S is a complete idempotent semiring s.t. $\top_{\leq_\mathrm{N}}$ exists), there is an ordinal α such that ${\mathrm{T_P}}^\alpha(\bot_\preceq)$ is the least fixpoint of $\mathrm{T_P}$, which coincides with $\mathrm{T_P}$'s least prefixpoint.*

Example 4. We illustrate $\mathrm{T_P}$ on the program from Example 1. The immediate consequence operator is instantiated as $\mathrm{T_P}(\mathrm{I})(H) = \min\{\sum_{i=1}^n \mathrm{I}(B_i) | H \text{ :- } B_1, \ldots, B_n \in P\}$. The bottom interpretation $\bot_\preceq$ for this semiring maps each semiring element to itself and each atom to the bottom element ∞.

In view of Corollary 1, we obtain $\mathrm{lfp}(\mathrm{T_P})$ stepwise as illustrated in Table 1. A fixpoint is reached at I_4, as $\mathrm{T_P}(\mathrm{I}_4) = \mathrm{I}_4$.

To show the equivalence between our model-theoretic semantics and our fixpoint semantics, we require that the prefixpoints of $\mathrm{T_P}$ coincide with the semiring models of P.

Lemma 11. *Let S be ordered by the complete lattice $\langle \mathcal{L}, \leq \rangle$. $\mathrm{I} \in IS_P$ is a semiring model of PSCLP(S) P iff $\mathrm{T_P}(\mathrm{I}) \preceq \mathrm{I}$.*

From here, we obtain the equivalence between the minimal semiring model of P and the least fixpoint of $\mathrm{T_P}$.

Lemma 12. *Let S be ordered by the complete lattice $\langle \mathcal{L}, \leq \rangle$. For PSCLP(S) P, $M_P^s = \mathrm{lfp}(\mathrm{T_P})$.*

5 Generalized Negation for Semiring-based Constraint Logic Programming

As presented up until this point, our formalism can only represent positive expressions. We can express that one may use mass transit if there *is* an available train connection, but we would have no good way to express that cycling is a good alternative if it is *not* raining. We now extend our formalism with a notion of negation so that negative expressions may also be represented. This notion of negation will follow the idea of negation as failure [Clark, 1977], where the negation of an atom may be derived if the atom itself is not derivable.

Definition 10. Given a semiring $S = \langle \mathcal{L}, +, \times, \mathbf{0}, \mathbf{1} \rangle$ and a set of atoms $\mathcal{A}$, *negated atoms* are defined as $\mathrm{not}\, a$ where $a \in \mathcal{A}$. Atoms or their negation are referred to as *literals*, and *generalized atoms* are literals or semiring elements. A *normal semiring-based constraint logic program* based on S (NSCLP(S) for short[6]) is a set of *normal clauses* of the form $H \text{ :- } B_1, \ldots, B_n$ where H, the *head*, is an atom and each B_i appearing in the (possibly empty) *body* is a generalized atom.

Normal logic programs, of course, interpret *true* and *false* as each other's negation. We generalize negation in normal logic programs with a notion of negation similar to that used in Gödel logics [Gödel, 1932], also used in the semiring-based formalism of Eiter and Kiesel [Eiter and Kiesel, 2020b]. Here we make use of $\mathbf{1}$ and $\mathbf{0}$, the neutral and absorbing elements of the multiplicative operator $\times$; if an atom is interpreted with $\mathbf{0}$—and it would thus nullify the interpretation of any conjunction it appears in—we interpret its negation as $\mathbf{1}$, leaving the interpretation of the further conjunction unaffected. Meanwhile, if an atom is interpreted as anything other than $\mathbf{0}$—and it would thus not nullify the interpretation of the conjunction it appears in—we interpret its negation as $\mathbf{0}$, nullifying the interpretation of any conjunction it appears in. Take, e.g. I_4 from Example 3. We have $\mathrm{I}_4(\mathrm{not}\ \mathtt{car(a)}) = \infty$, as $\mathrm{I}_4(\mathtt{car(a)}) = 3 \neq \infty$. Meanwhile, for `rain(a)`—an atom without a clause defining it—we have $\mathrm{I}_4(\mathrm{not}\ \mathtt{rain(a)}) = 0$, as $\mathrm{I}_4(\mathtt{rain(a)}) = \infty$. This notion of negation generalizes negation as known from nlps while making only minimal assumptions on the semiring.

Note. A semiring permits only a single absorbing element for $\times$—namely $\mathbf{0}$; for any $\times$-absorbing element a, we have that $a \times \mathbf{0} = a = \mathbf{0} = \mathbf{0} \times a$.

Definition 11. An *interpretation* I (Def. 4) is extended to negated literals by $\mathrm{I}(\mathrm{not}\, a) = \begin{cases} \mathbf{1} & \text{if } \mathrm{I}(a) = \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases}$.

As is well-known from normal logic programs, introducing negation leads to non-monotonicity of the $\mathrm{T_P}$-operator:

Example 5. Consider the NSCLP($\mathbb{B}$) $P = \{a \text{ :- } \mathrm{not}\, b\}$, where *true* $>$ *false* as usual. Now consider two interpretations $\mathrm{I}_1 = \{a : \textit{true}, b : \textit{true}\}$ and $\mathrm{I}_2 = \{b : \textit{false}, b : \textit{false}\}$. We see that $\mathrm{I}_2 \preceq \mathrm{I}_1$. Applying $\mathrm{T_P}$ we get $\mathrm{T_P}(\mathrm{I}_1)(a) = \mathrm{I}_1(\mathrm{not}\, b) = \textit{false}$, $\mathrm{T_P}(\mathrm{I}_2)(a) = \mathrm{I}_2(\mathrm{not}\, b) = \textit{true}$, and $\mathrm{T_P}(\mathrm{I}_1)(b) = \textit{false} = \mathrm{T_P}(\mathrm{I}_2)(b)$. Note that, because there is no predicate defining b, the value assigned by $\mathrm{T_P}$ to b is *false*—the neutral element for $\vee$ resulting from summation over an empty set—under either interpretation. Comparing the values assigned to a and b now shows that $\mathrm{I}_2 \preceq \mathrm{I}_1$ but $\mathrm{T_P}(\mathrm{I}_2) \not\preceq \mathrm{T_P}(\mathrm{I}_1)$.

Naturally, this behaviour means that previous results relying on the assumed $\leq$-monotonicity of $+$ and $\times$ cannot be

[6] We write NSCLP when the semiring is clear or unimportant.

presumed to hold for NSCLPs. Most importantly, the non-monotonicity of $\mathrm{T_P}$ means that we can no longer use Theorem 1 to construct its least fixpoint. In the following sections, we will circumvent this behaviour by means of AFT.

6 Normal Semiring-based Constraint Logic Programming in Approximation Fixpoint Theory

In the previous section, we saw that we can no longer derive the monotonicity of our immediate consequence operator from the monotonicity of $+$ and $\times$. This left us without a constructive definition of the least fixpoint semantics for our program. In this section, we capture our framework in AFT by approximating the immediate consequence operator, thus bestowing NSCLPs with the semantic notions recalled in Section 2.2.

Note. Hereafter, unless otherwise specified, we assume that $\langle \mathcal{L}, \leq \rangle$ is a complete lattice. Note that, by extension, $\langle IS_P, \preceq \rangle$ is assumed to be a complete lattice as well.

6.1 Approximating the Immediate Consequence Operator

We first consider the most precise, *ultimate*, approximator $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}$ of the immediate consequence operator $\mathrm{T_P}$.

Definition 12. $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}(\mathrm{I}_1, \mathrm{I}_2) := (\bigwedge \mathrm{T_P}([\mathrm{I}_1, \mathrm{I}_2]), \bigvee \mathrm{T_P}([\mathrm{I}_1, \mathrm{I}_2]))$ for $\mathrm{I}_1, \mathrm{I}_2 \in IS_P^c$.

While the ultimate approximator is the most precise approximator, its evaluation is rather costly—requiring the evaluation of $\mathrm{T_P}$ for every interpretation in the input approximation, prior to identifying the glb and lub. This typically results in a higher computational complexity [Denecker *et al.*, 2004]. For this reason we also introduce another approximator which is not necessarily as precise, but is more economical to use.

For this second approximator, we generalize the four-valued operator originally introduced by [Fitting, 2002]. Recall that consistent approximations $(\mathrm{I}_l, \mathrm{I}_u) \in IS_P^c$ have a lower bound I_l—which is most conservative in its interpretation—and an upper bound I_u—which is most generous in its interpretations. In other words, for any atom $A \in \mathcal{A}_P$, $\mathrm{I}_l(A)$ is the lowest value of any $\mathrm{I}(A)$ with $\mathrm{I} \in [\mathrm{I}_l, \mathrm{I}_u]$ and $\mathrm{I}_u(a)$ the highest. Now, to approximate $\mathrm{T_P}$, we find a new lower bound taking the same sum-product as we would when evaluating $\mathrm{T_P}$ but—instead of considering a single interpretation—we evaluate generalized atoms without negation using I_l (yielding the most conservative values) and evaluate negated atoms using I_u (nullifying whenever possible). The new upper bound is found the same way, switching I_l and I_u. This way of approximating $\mathrm{T_P}$ relies on the additional assumption that $\mathbf{0}$ lies at the bottom of our semiring ordering, such that nullifying an expression cannot increase its value; this way, using the given upper bound to evaluate negated literals results in lesser values and vice versa.

Definition 13. Let $\mathrm{I}_1, \mathrm{I}_2 \in IS_P$; $H \in \mathcal{A}_P$; $\mathcal{T}_{\mathrm{P}_1}^{\mathrm{f}}(\mathrm{I}_1, \mathrm{I}_2)(H) = \sum\{\prod_{i=1}^{n} \mathrm{I}'(\mathrm{I}_1, \mathrm{I}_2)(B_i) | H \text{ :- } B_1, \ldots, B_n \in P\}$; and $\mathrm{I}'(\mathrm{I}_1, \mathrm{I}_2)(B_i) = \begin{cases} I_2(B_i) & \text{if } B_i \text{ is of the form not } F \\ I_1(B_i) & \text{otherwise} \end{cases}$.

We define $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(\mathrm{I}_1, \mathrm{I}_2) = (\mathrm{I}_3, \mathrm{I}_4)$ s.t. for all $A \in \mathcal{A}_P$, $\mathrm{I}_3(A) = \mathcal{T}_{\mathrm{P}_1}^{\mathrm{f}}(\mathrm{I}_1, \mathrm{I}_2)(A)$ and $\mathrm{I}_4(A) = \mathcal{T}_{\mathrm{P}_1}^{\mathrm{f}}(\mathrm{I}_2, \mathrm{I}_1)(A)$.

It follows immediately from the results from [Denecker *et al.*, 2004, Theorem 5.6] that $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}$ is an approximator of $\mathrm{T_P}$.

Somewhat surprisingly, $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ is not generally $\leq_{\mathrm{p}}$-monotone:

Example 6. We present an example wherein $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ is not $\leq_{\mathrm{p}}$-monotone.

Take the semiring $\mathbb{Z} = \langle \mathbb{Z} \cup \{-\infty, +\infty\}, +, \cdot, 0, 1 \rangle$ consisting of the integers with positive and negative infinity, ordered by $\leq$ as usual. Consider an NSCLP($\mathbb{Z}$) consisting of a single clause p :- not q, r. Now consider two approximations $A_1 = (\mathrm{I}_1, \mathrm{I}_2)$ and $A_2 = (\mathrm{I}_1, \mathrm{I}_3)$ where $\mathrm{I}_1 = \{p : -1, q : -1, r : -1\}$, $\mathrm{I}_2 = \{p : 1, q : 1, r : 1\}$, and $\mathrm{I}_3 = \{p : 0, q : 0, r : 0\}$. We see that $A_1 \leq_{\mathrm{p}} A_2$. We now apply $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ to A_1 and A_2 and get $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_1) = (\{p : 0, q : 0, r : 0\}, \{p : 0, q : 0, r : 0\})$ and $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_2) = (\{p : -1, q : 0, r : 0\}, \{p : 0, q : 0, r : 0\})$. Even though A_2 is more precise than A_1, we do not have that $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_1) \leq_{\mathrm{p}} \mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_2)$; in fact, $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_1)$ is strictly more precise than $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}(A_2)$. $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ is not $\leq_{\mathrm{p}}$-monotone for this NSCLP. The reason for this failure of $\leq_{\mathrm{p}}$-monotonicity is that the upper bound of A_2 being lower (i.e. more precise) than that of A_1 (namely 0 instead of 1 for every atom) causes not q in the rule body to evaluate to 1, which in turn "allows" the truth value of r to determine the truth value of p. However, as this "positive" influence of r means that p attains the negative value $\mathrm{I}_1(r) = -1$, we obtain a new lower bound for p that is negative and thus lower than the new lower bound obtained in view of A_1. Thus, $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ will not behave as an approximator when applied to semirings that have elements below the $\mathbf{0}$-element.

Working out the same example for $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}$, we get $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}(A_1) = (\{p : -1, q : 0, r : 0\}, \{p : 1, q : 0, r : 0\})$ and $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}(A_2) = (\{p : -1, q : 0, r : 0\}, \{p : 0, q : 0, r : 0\})$. Here we do have that $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}(A_1) \leq_{\mathrm{p}} \mathcal{T}_{\mathrm{P}}^{\mathrm{u}}(A_2)$. This is no coincidence, as $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}}$ is guaranteed to be an approximator [Denecker *et al.*, 2004].

Whenever the additive and multiplicative operators are monotone and $\mathbf{0}$ is the minimum element of the semiring ordering, $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ is an approximator of $\mathrm{T_P}$:

Lemma 13. *$\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ is an approximator of $\mathrm{T_P}$ if and only if S is positively ordered by the complete lattice $\langle \mathcal{L}, \leq \rangle$.*

We can now construct the KK fixpoints of our two approximators, to approximate all the fixpoints of $\mathrm{T_P}$, thus generalizing the semantics from PSCLPs to NSCLPs:

Corollary 2. *There is an ordinal α such that $\mathcal{T}_{\mathrm{P}}^{\mathrm{u}\alpha}(\bot_{\leq_{\mathrm{p}}}) = \mathrm{lfp}_{\leq_{\mathrm{p}}}(\mathcal{T}_{\mathrm{P}}^{\mathrm{u}})$. If S is positively ordered by the complete lattice $\langle \mathcal{L}, \leq \rangle$, the same holds for $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$.*

Note. If $\mathrm{lfp}_{\leq_{\mathrm{p}}}(\mathcal{T}_{\mathrm{P}}^{\mathrm{u}})$ and $\mathrm{lfp}_{\leq_{\mathrm{p}}}(\mathcal{T}_{\mathrm{P}}^{\mathrm{f}})$ are exact, $\mathrm{T_P}$ has only a single fixpoint. This is the case in particular if P is positive.

Example 7. We illustrate the use of an approximator by constructing the KK fixpoint of $\mathcal{T}_{\mathrm{P}}^{\mathrm{f}}$ for our running example, extended with three additional clauses to incorporate negation. That is, $P = \{c_1, \ldots, c_{10}\}$ with:

```
c8: solution(a) :- path(a,d).
c9: path(a,d) :- bicycle(a).
c10: bicycle(a) :- 1, not rain(a).
```

$\mathcal{T}_{\mathrm{P}_1}^{\mathrm{f}}$ is instantiated as $\mathcal{T}_{\mathrm{P}_1}^{\mathrm{f}}(\mathrm{I}_1, \mathrm{I}_2)(H) = \min\{\sum_{i=1}^{n} \mathrm{I}'(\mathrm{I}_1, \mathrm{I}_2)(B_i) | H \text{ :- } B_1, \ldots, B_n \in P\}$. Table 2

	$\bot_{\leq_\mathrm{p}}$	A_1	A_2	A_3	A_4		$\bot_{\leq_\mathrm{p}}$	A_1	A_2	A_3	A_4
`rain(a)`	$(\infty,0)$	(∞,∞)	(∞,∞)	(∞,∞)	(∞,∞)	`train(a)`	$(\infty,0)$	$(2,2)$	$(2,2)$	$(2,2)$	$(2,2)$
`car(a)`	$(\infty,0)$	$(3,3)$	$(3,3)$	$(3,3)$	$(3,3)$	`bicycle(a)`	$(\infty,0)$	$(\infty,1)$	$(1,1)$	$(1,1)$	$(1,1)$
`mass_transit(a)`	$(\infty,0)$	$(\infty,0)$	$(2,2)$	$(2,2)$	$(2,2)$	`path(a,d)`	$(\infty,0)$	$(\infty,0)$	$(\infty,1)$	$(1,1)$	$(1,1)$
`path(a,c)`	$(\infty,0)$	$(\infty,0)$	$(3,3)$	$(3,3)$	$(3,3)$	`path(a,b)`	$(\infty,0)$	$(\infty,0)$	$(\infty,0)$	$(2,2)$	$(2,2)$
`solution(a)`	$(\infty,0)$	$(\infty,0)$	$(\infty,0)$	$(3,0)$	$(1,1)$						

Table 2: Application of $\mathcal{T}_\mathrm{P}^\mathrm{f}$ for Example 7, starting from $\bot_{\leq_\mathrm{p}}$.

shows the application of the corresponding $\mathcal{T}_\mathrm{P}^\mathrm{f}$, starting from $\bot_{\leq_\mathrm{p}}$, to construct the KK fixpoint A_4.

6.2 Stable Semantics for Normal Semiring-based Constraint Logic Programs

KK fixpoints of $\mathcal{T}_\mathrm{P}^\mathrm{u}$ and $\mathcal{T}_\mathrm{P}^\mathrm{f}$ approximate all fixpoints of $\mathrm{T_P}$, but might sanction self-supporting reasoning:

Example 8. Consider an NSCLP($\mathbb{B}$) P consisting of two clauses: $p\,\mathtt{:-}\,\mathrm{not}\,q$ and $q\,\mathtt{:-}\,q$. Applying $\mathcal{T}_\mathrm{P}^\mathrm{f}$ gives us $\mathcal{T}_\mathrm{P}^\mathrm{f}(\bot_{\leq_\mathrm{p}}) = (\{p : \mathit{false}, q : \mathit{false}\}, \{p : \mathit{true}, q : \mathit{true}\}) = \bot_{\leq_\mathrm{p}}$; thus $\bot_{\leq_\mathrm{p}}$ is the KK fixpoint of $\mathcal{T}_\mathrm{P}^\mathrm{f}$, which means that q stays undecided because of the self-supporting rule $q\,\mathtt{:-}\,q$, which also keeps p undecided (in view of $p\,\mathtt{:-}\,\mathrm{not}\,q$).

The only reason for making q *true* in the upper bound of the KK fixpoint in the example above is the fact that q was made *true* in the upper bound of the least precise approximation. This type of "because-I-said-so" reasoning is avoided by using the stable operator and its $\leq_\mathrm{p}$-least fixpoint—the WF fixpoint, which we study in this section.

We first observe these operators are well-defined:

Corollary 3. *For any* $(\mathrm{I}_1,\mathrm{I}_2) \in IS_P^c$ *there are ordinals* α *and* β *such that* $(\bigwedge \mathrm{T_P}([\cdot,\mathrm{I}_2]))^\alpha(\bot_\preceq) = \mathrm{lfp}(\mathcal{T}_{\mathrm{P}\,l}^\mathrm{u}(\cdot,\mathrm{I}_2))$, $(\bigvee \mathrm{T_P}([\mathrm{I}_1,\cdot]))^\beta(\bot_\preceq) = \mathrm{lfp}(\mathcal{T}_{\mathrm{P}\,u}^\mathrm{u}(\mathrm{I}_1,\cdot))$, *and the* stable operator *for* $\mathcal{T}_\mathrm{P}^\mathrm{u}$ *is* $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{u})(\mathrm{I}_1,\mathrm{I}_2) = (\mathrm{lfp}(\mathcal{T}_{\mathrm{P}\,l}^\mathrm{u}(\cdot,\mathrm{I}_2)), \mathrm{lfp}(\mathcal{T}_{\mathrm{P}\,u}^\mathrm{u}(\mathrm{I}_1,\cdot)))$. *The same holds for* $\mathcal{T}_\mathrm{P}^\mathrm{f}$ *if* S *is positively ordered by* $\langle\mathcal{L},\leq\rangle$.

The fixpoints of these stable operators, called the stable fixpoints of $\mathcal{T}_\mathrm{P}^\mathrm{u}$ and $\mathcal{T}_\mathrm{P}^\mathrm{f}$ respectively, are $\leq_\mathrm{v}$-minimal fixpoints of their respective bilattice operator (this is an immediate corollary of Theorem 4 of [Denecker *et al.*, 2001]) and they generalize the WF and stable model semantics known from normal logic programs (see Theorem 2).

We can now also construct the well-founded fixpoints of our two stable operators. These fixpoints approximate all stable fixpoints of their respective bilattice operators.

Corollary 4. *There is an ordinal* α *such that* $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{u})^\alpha(\bot_{\leq_\mathrm{p}}) = \mathrm{lfp}_{\leq_\mathrm{p}}(\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{u}))$. *If* S *is positively ordered by* $\langle\mathcal{L},\leq\rangle$, *the same is true for* $\mathcal{T}_\mathrm{P}^\mathrm{f}$.

Example 9. To illustrate the use of the stable operator, we apply $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{f})$ to the program presented in Example 8 We first construct the WF fixpoint, by repeatedly applying $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{f})$, starting from $\bot_{\leq_\mathrm{p}}$. Denoting interpretations as sets of *true* atoms, we get the following. First, $\bot_{\leq_\mathrm{p}} = (\emptyset,\{p,q\})$. Then, $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{f})((\emptyset,\{p,q\})) = (\{p\},\{p\})$ since $\mathrm{lfp}(\mathcal{T}_{\mathrm{P}1}^\mathrm{f}(\cdot,\{p,q\})) = \{p\}$ and $\mathrm{lfp}(\mathcal{T}_{\mathrm{P}1}^\mathrm{f}(\cdot,\emptyset)) = \{p\}$. Finally, $\mathcal{S}(\mathcal{T}_\mathrm{P}^\mathrm{f})(\{p\},\{p\}) = (\{p\},\{p\})$ so $(\{p\},\{p\})$ is the WF fixpoint of $\mathcal{T}_\mathrm{P}^\mathrm{f}$. Since this WF fixpoint is exact, we also know it to be the only stable fixpoint of $\mathcal{T}_\mathrm{P}^\mathrm{f}$.

Finally, since our operators generalize those known from normal logic programming, and AFT has been shown to faithfully capture all the main semantics from nlp [Pelov *et al.*, 2007], we faithfully generalize these main semantics:

Theorem 2. *Given an NSCLP($\mathbb{B}$), the following hold: (1)* $(\mathrm{I}_1,\mathrm{I}_2)$ *is a stable fixpoint of* $\mathcal{T}_\mathrm{P}^\mathrm{f}$ *iff it is a partial stable model according to [Przymusinski, 1990]; (2)* $(\mathrm{I}_1,\mathrm{I}_2)$ *is the WF fixpoint of* $\mathcal{T}_\mathrm{P}^\mathrm{f}$ *iff it is the WF model according to [Przymusinski, 1990]; (3)* $(\mathrm{I}_1,\mathrm{I}_2)$ *is a stable fixpoint of* $\mathcal{T}_\mathrm{P}^\mathrm{u}$ *iff it is a partial stable model according to [Denecker* et al.*, 2004]; (4)* $(\mathrm{I}_1,\mathrm{I}_2)$ *is the WF fixpoint of* $\mathcal{T}_\mathrm{P}^\mathrm{u}$ *iff it is the WF model according to [Denecker* et al.*, 2004].*

7 Conclusion, in view of Related Work

We have unified existing semantics [Bistarelli *et al.*, 2001; Khamis *et al.*, 2023] for PSCLPs in this paper, and generalized the syntax and semantics to allow for default negation on the basis of AFT, resulting in a family of well-behaved semantics that also generalizes the well-known semantics for nlps. To our best knowledge only a few other approaches combine semirings and logic programming, which we discuss now.

Firstly, Eiter and Kiesel 2020a provide a framework that captures many approaches to answer set programs taking into account algebraic constraints. In this work, terms (like `a` in `train(a)`) are allowed to be interpreted using semiring values and operations. However, while terms are assigned semiring values, formulas are still interpreted in a discrete way, by assigning the neutral element of $\times$ if it is true while assigning the neutral element of the $+$ if it is true. Thus, they use a different interpretation of formulas, where semirings are seen as oracles to evaluate the constraints. [Eiter and Kiesel, 2020a] show that a host of different systems, e.g. integrations of satisfaction modulo theories into ASP [Cabalar *et al.*, 2020b; Cabalar *et al.*, 2020a], can be captured within their framework, and the differences described above also apply to such systems.

Secondly, there is work combining logic programs with semirings in generalizations of model-counting [Derkinderen *et al.*, 2024; Kimmig *et al.*, 2017]. Such works differ from ours in that their semantics are obtained by assigning every discrete or Boolean interpretation of a logic program a semiring value, generalizing the ProbLog semantics [Raedt *et al.*, 2007], whereas we use semiring values to build our interpretations.

Future work includes 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].

References

[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.

[Blyth, 2005] T.S. Blyth. *Lattices and Ordered Algebraic Structures*. Universitext. Springer-Verlag, London, 2005.

[Cabalar *et al.*, 2020a] Pedro Cabalar, Jorge Fandinno, Torsten Schaub, and Philipp Wanko. An ASP Semantics for Constraints Involving Conditional Aggregates. 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 664–671, 2020.

[Cabalar *et al.*, 2020b] Pedro Cabalar, Jorge Fandinno, Torsten Schaub, and Philipp Wanko. A Uniform Treatment of Aggregates and Constraints in Hybrid ASP. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020.*, pages 193–202, 2020.

[Ceri *et al.*, 1989] Stefano Ceri, Georg Gottlob, and Letizia Tanca. What you Always Wanted to Know About Datalog (And Never Dared to Ask). *IEEE Trans. Knowl. Data Eng.*, 1(1):146–166, 1989.

[Clark, 1977] Keith L. Clark. Negation as Failure. In *Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de Recherches de Toulouse, France, 1977.*, pages 293–322, 1977.

[Cousot and Cousot, 1979] Patrick Cousot and Radhia Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 82(1):43–57, May 1979.

[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.

[Denecker *et al.*, 2004] Marc Denecker, Victor W. Marek, and Miroslaw Truszczynski. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Inf. Comput.*, 192(1):84–121, 2004.

[Derkinderen *et al.*, 2024] Vincent Derkinderen, Robin Manhaeve, Pedro Zuidberg Dos Martires, and Luc De Raedt. Semirings for probabilistic and neuro-symbolic logic programming. *Int. J. Approx. Reason.*, 171:109130, 2024.

[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.

[Fitting, 2002] Melvin Fitting. Fixpoint semantics for logic programming a survey. *Theor. Comput. Sci.*, 278(1-2):25–51, 2002.

[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.

[Hannula, 2023] Miika Hannula. Conditional independence on semiring relations. *CoRR*, abs/2310.01910, 2023.

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

[Lloyd, 1987] John W. Lloyd. *Foundations of Logic Programming, 2nd Edition*. Springer, 1987.

[Pelov *et al.*, 2007] Nikolay Pelov, Marc Denecker, and Maurice Bruynooghe. Well-founded and stable semantics of logic programs with aggregates. *Theory Pract. Log. Program.*, 7(3):301–353, 2007.

[Przymusinski, 1990] Teodor C. Przymusinski. The Well-Founded Semantics Coincides with the Three-Valued Stable Semantics. *Fundam. Inform.*, 13(4):445–463, 1990.

[Raedt *et al.*, 2007] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467, 2007.

[Tarski, 1955] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, June 1955.

[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.