Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Complexity of Logics with Semiring Semantics

Timon Barlag[1] , **Nicolas Fröhlich**[1] , **Teemu Hankala**[2] , **Miika Hannula**[2,3] ,
Minna Hirvonen[1] , **Vivian Holzapfel**[1] , **Juha Kontinen**[2] , **Arne Meier**[1] , **Laura Strieker**[1]
[1]Leibniz Universität Hannover, Hannover, Germany
[2]University of Helsinki, Helsinki, Finland
[3]University of Tartu, Tartu, Estonia
{barlag,nicolas.froehlich,minna.hirvonen,holzapfel,meier,strieker}@thi.uni-hannover.de,
{juha.kontinen,teemu.hankala}@helsinki.fi, miika.hannula@ut.ee

Abstract

We study the expressive power and computational properties of first-order logic and its extensions under the semiring semantics originating from the seminal work of Green, Karvounarakis, and Tannen. While semiring semantics is currently extensively used, e.g., in the study of provenance in database theory and description logic, a comprehensive computational analysis of these logics acting over general semirings is still lacking. We analyse expressivity, and complexity of model-checking of first-order formulae in this framework, providing characterizations in terms of generalized Blum–Shub–Smale machines over semirings. We also show a variant of Fagin's theorem, i.e., a logical characterization of non-deterministic polynomial time over semirings using a version of existential second-order logic. We further generalize Cook's theorem for the semiring framework and show that propositional satisfiability in the semiring semantics is complete for this notion of NP, and that the true existential first-order theory of the semiring is complete for its Boolean fragment.

1 Introduction

Over the last two decades, the use of semiring semantics to study provenance for database query languages has become a very active area of research (Green, Karvounarakis, and Tannen 2007; Tannen 2017; Green and Tannen 2017; Khamis et al. 2024). More recently, the use of semiring semantics has been also considered in relation to other database tasks such as consistent query answering (Kolaitis et al. 2025; Fröhlich et al. 2025). Furthermore, provenance analysis has been also studied for description logics (see (Calvanese et al. 2019; Bourgaux and Ozaki 2019; Bourgaux, Ozaki, and Peñaloza 2023) and the references therein), and for full first-order logic and its extensions (Grädel and Tannen 2017).

Semirings are algebraic structures that generalize rings by relaxing the requirement for additive inverses, and they have found numerous applications in computer science and AI due to their versatility and modularity in modelling and analysing computational problems (Peeva 1991; Gaubert and Katz 2006; Litvinov et al. 2013; Möller 2013; Esparza, Luttenberger, and Schlund 2015; Vrana 2022; Eiter and Kiesel 2023). Different choices of semirings correspond to different computational paradigms or problem domains. Classical decision problems are typically studied over the Boolean semiring $\mathbb{B}$, counting problems over the semiring of natural numbers $\mathbb{N}$, and problems involving geometric or continuous aspects over the semiring of non-negative reals $\mathbb{R}_{\geq 0}$. Tropical semirings, on the other hand, have applications in performance analysis (Omanovic, Oblak, and Curk 2023) and reachability problems (Gaubert and Katz 2006).

Several computational models admit natural generalizations to semirings. Weighted automata and weighted Turing machines assign semiring elements to transitions, representing quantities such as probabilities, costs, or capacities, and recognize formal power series $r\colon \Sigma^* \to K$ over a finite alphabet Σ and a semiring K (see, e.g., (Kostolányi 2024; Badia et al. 2024; Eiter and Kiesel 2023)). Conversely, languages over the reals and, more generally, over rings have been investigated using register machines that operate directly on numerical values. The Blum–Shub–Smale (BSS) machine model (Blum, Shub, and Smale 1988; Preparata and Shamos 1985) provides a prominent example and extends naturally to semirings and arbitrary first-order structures (Poizat 1995; Bournez et al. 2006). Unlike classical Turing machines, BSS machines treat elements of the underlying semiring as atomic objects: each element occupies a single cell regardless of its representation size, allowing, for instance, transcendental numbers to be manipulated in constant time.

Nowadays semirings play a central role in database theory research through semiring provenance. In this approach, query evaluation yields more than a Boolean answer by annotating atomic facts in database tables with elements of a semiring K and propagating these annotations through the query (Green, Karvounarakis, and Tannen 2007). Depending on the chosen semiring, provenance information can encode information on the confidence, cost, or the number of satisfying assignments of a query (see, e.g., (Calautti et al. 2024; Im et al. 2024; Eldar, Carmeli, and Kimelfeld 2024; Muñoz Serrano, Riveros, and Vansummeren 2024; ten Cate et al. 2024)). On the other hand, K-relations (or weighted structures) provide a unifying abstraction that subsumes, for example, bags and probability distributions, with applications extending beyond database theory (cf. (Atserias and Kolaitis 2025; Grohe et al. 2025)).

Semiring semantics has also been defined for first-order logic (Grädel and Tannen 2017; Tannen 2017). In this setting, it has been studied via central themes of finite model theory,

including Ehrenfeucht–Fraïssé games, locality, 0–1 laws, and definability up to isomorphism (Bizière, Grädel, and Naaf 2023; Brinke, Grädel, and Mrkonjic 2024; Grädel et al. 2022; Grädel and Mrkonjic 2021). The framework has since been extended to more expressive logics, such as fixed-point logic (Dannert et al. 2021) and team-based logics (Barlag et al. 2023). Despite their growing popularity and range of applications, the complexity-theoretic properties of logics under semiring semantics remain largely unexplored. A systematic investigation of these properties naturally relies on models of algebraic computation, rather than on classical complexity theory.

From a complexity-theoretic perspective, BSS machines provide a natural computation model for logics acting on semirings. It is worth noting that logical characterizations for algebraic complexity classes over the reals defined by BSS machines are known in the context of metafinite model theory (Grädel and Meer 1995) and similar results are also known for BSS machines computing over arbitrary first-order structures (de Naurois 2004; Kirn et al. 2025). However, these results cannot be directly utilized to analyse the complexity of logics in the semirings semantics due to the differences between the two semantical frameworks.

Contributions. The complexity results in this article are based on a version of the BSS machine computing over a semiring K. We use these machines for computing functions $f\colon K^* \to K$ and recognizing languages $L \subseteq K^*$. On the logical side, we consider versions of propositional, first-order and existential second-order logic interpreted via commutative and positive semirings. Our languages extend the standard systems of these logics by a formula value comparison $\phi = \psi$ that generalizes identity checks from variables to arbitrary formulae. We demonstrate the effects of adding this construct to first-order logic which is very useful in expressing conditionals.

We analyse the expressivity and complexity of model checking of first-order logic formulae using the BSS machine model showing that, analogously to the Boolean setting, the model checking problem is in PSPACE_K and data complexity is in P_K. For existential second-order logic and non-deterministic computations we establish analogues of Fagin's and Cook's theorems in the BSS model. In particular, we show that the satisfiability problem of propositional logic under semiring semantics is complete for this notion of NP, and that the true existential first-order theory of the semiring is complete for its Boolean fragment. What is more, we show that the polynomial-time reductions for these completeness results can be carried out in a manner that does not depend on computing the arithmetic operations of the semiring.

Related work. Research on logics acting on semirings and other algebraic structures is currently active on several fronts, hence we discuss the similarities and differences of our work compared to some of the other approaches. First of all, although the BSS machine model was initially defined for rings in (Blum, Shub, and Smale 1988; Preparata and Shamos 1985), it has been also generalized for arbitrary first-order structures (Poizat 1995). Hence, our version for semirings is not novel in that sense. Furthermore, already the work (Goode 1994) shows a version of Corollary 4 for BSS machines for a general class of first-order structures with all constants, and (Bürgisser and Cucker 2003) shows a result similar to Corollary 5 for Boolean constants and ordered rings. We also want to note that versions of Fagin's theorem for BSS machines over reals and also general first-order structures have been shown in (Grädel and Meer 1995; de Naurois 2004; Kirn et al. 2025) in the context of metafinite logic. There are also several characterizations of NP for weighted Turing machines (Kostolányi 2024; Badia et al. 2024; Eiter and Kiesel 2023; Badia et al. 2025) using logics under semiring semantics.

Organization. In Section 2, we review basic notions on semirings, state our background assumptions and introduce the BSS machine model for semirings. In Section 3, we show that the satisfiability problem of propositional logic under the semiring semantics is a complete problem for NP_K. In Section 4, we consider first-order logic and its extension by the formula value comparison $\phi = \psi$ and their model checking problem. In Section 5, We define a version of existential second-order logic ($\mathrm{ESO}^{=}_{K}$) in the semiring framework and show a version of Fagin's theorem. Finally, in Section 6, we consider the existential first-order theory for K (under the usual Tarski's semantics), and show using novel non-arithmetic reductions that it is a complete problem for the Boolean part of the restriction of NP_K to machines that use only the values 0 and 1 as machine constants. In fact, we show a generalized version of this result for any set $X \supseteq \{0, 1\}$ of elements of K treated as constant symbols.

2 Preliminaries

We assume familiarity with basic concepts in theoretical computer science, e.g., complexity theory (Sipser 1997). We start with the fundamental definition of a semiring.

Definition 1. A *semiring* is a tuple $K = (K, +, \cdot, 0, 1)$, where $+$ and $\cdot$ are binary operations on K, $(K, +, 0)$ is a commutative monoid with identity element 0, $(K, \cdot, 1)$ is a monoid with identity element 1, $\cdot$ left and right distributes over $+$, and $x \cdot 0 = 0 = 0 \cdot x$ for all $x \in K$. K is called *commutative* if $(K, \cdot, 1)$ is a commutative monoid. As usual, we often write ab instead of $a \cdot b$. A semiring is *ordered* if there exists a partial order $\leq$ (meaning $\leq$ is reflexive, transitive, and antisymmetric) such that $0 \leq 1$, and for all $a, b, c \in K$ $a \leq b$ implies $a + c \leq b + c$, and $a \leq b \wedge 0 \leq c$ implies $ac \leq bc \wedge ca \leq cb$. A semiring is *positive* if it has no divisors of 0, i. e., $ab \neq 0$ for all $a, b \in K$, where $a \neq 0 \neq b$, and if $a + b = 0$ implies that $a = b = 0$.

Remark 1. Throughout this paper, we consider semirings that are commutative, positive, and non-trivial (i. e., $0 \neq 1$). These assumptions are made for the sake of simplicity, even though the results in Section 6 require these assumptions only for non-triviality.

BSS machines were introduced as a model for real computation by (Blum, Shub, and Smale 1988) and were originally

defined over rings. The computation nodes of a BSS machine are usually formulated in terms of polynomial functions with coefficients from the underlying ring. In case this ring is actually a field, also rational functions, i. e., divisions of polynomials, are permitted. This model is modified to work over semirings K and called a BSS_K machine.

In BSS machines, the analogue of a Turing machine's tape is a state space consisting of infinite tuples with a dedicated centre.

Definition 2. For a set A, we let $A^* = \bigcup_{n\geq 0} A^n$. By A_* we denote the set $A^{\mathbb{Z}}$ of all $x = (\dots, x_{-2}, x_{-1}, x_0.x_1, x_2, \dots)$, where "." marks the first position. We define two shift operations on A_*: shift left σ_ℓ, where $\sigma_\ell(x_i) = x_{i+1}$, and its inverse, shift right σ_r, where $\sigma_r(x_i) = x_{i-1}$, for all $i \in \mathbb{Z}$.

Based on this, we define our BSS machine model of computation over semirings.

Definition 3 (BSS_K machines)**.** A BSS_K machine M consists of an input space $\mathcal{I} = K^*$, a state space $\mathcal{S} = K_*$ and an output space $\mathcal{O} = K^*$, together with a connected directed graph whose nodes are labelled by $1, \dots, N$. The nodes are of five different types as follows.

Input node: The node labelled by 1 is the only input node. The node is associated with a next node $\beta(1)$ and the input mapping $g_I \colon \mathcal{I} \to \mathcal{S}$.

Output node: The node labelled by N is the only output node. This node is not associated with any next node. Once this node is reached, the computation halts, and the result of the computation is placed on the output space by the output mapping $g_O \colon \mathcal{S} \to \mathcal{O}$.

Computation nodes: A computation node m is associated with a next node $\beta(m)$ and a mapping $g_m \colon \mathcal{S} \to \mathcal{S}$ such that for some $c \in K$ and $i, j, k \in \mathbb{Z}$ the mapping g_m is identity on all coordinates different from i and on current coordinate i one of the following holds:

- $g_m(x)_i = x_j + x_k$ (addition),
- $g_m(x)_i = x_j \cdot x_k$ (multiplication),
- $g_m(x)_i = c$ (constant assignment).

Branch nodes: A branch node m is associated with nodes $\beta^-(m)$ and $\beta^+(m)$. Given $x \in \mathcal{S}$ the next node is $\beta^-(m)$ if $x_1 = x_2$ and $\beta^+(m)$, otherwise. If K is ordered, instead the next node is $\beta^-(m)$ if $x_1 \leq x_2$ and $\beta^+(m)$, otherwise.

Shift nodes: A shift node m is associated either with shift left σ_ℓ or shift right σ_r, and a next node $\beta(m)$.

The *input mapping* $g_I \colon \mathcal{I} \to \mathcal{S}$ places an input $(x_1, \dots, x_n)$ in the state

$$(\dots, 0, \underbrace{1, \dots, 1}_{n}, 0.x_1, \dots, x_n, 0, \dots) \in \mathcal{S},$$

where the size of the input n is encoded in unary in the n first negative coordinates. The *output mapping* $g_O \colon \mathcal{S} \to \mathcal{O}$ maps a state to the string consisting of its first ℓ positive coordinates, where ℓ is the number of consecutive ones stored in the negative coordinates starting from the first negative coordinate. For instance,

$$g_O((\dots, 2, 1, 1, 1, 0.x_1, x_2, x_3, x_4, \dots)) = (x_1, x_2, x_3).$$

A *configuration* at any timestep of the computation consists of a node $m \in \{1, \dots, N\}$ and a current state $x \in \mathcal{S}$. The (sometimes partial) *input-output function* $f_M \colon K^* \to K^*$ of a machine M is now defined in the obvious manner. Note that for finite computations, that means that there are only ever finitely many nonzero coordinates in the state space. A function $f \colon K^* \to K^*$ is *computable* if $f = f_M$ for some machine M. A language $L \subseteq K^*$ is *decided* by a BSS_K machine M if

$$\bar{x} \in L \quad \text{if and only if} \quad f_M(\bar{x}) \neq 0,$$

holds for all $\bar{x} \in K^*$.

The constant values of the semiring that are used in the computation nodes corresponding to constant assignments in the previous definition are referred to as *machine constants*. Since the number of computation nodes of a BSS_K machine is finite, each machine has a only finite number of machine constants.

Arbitrary computations and comparisons need a way to remember intermediate values. To this end, we use $(\dots, 0.\, x_1, 1, x_2, 1 \dots, x_n, 1, 0 \dots)$ as a more useful starting state of a BSS_K machine. The gap between two input values allows the machine to temporarily store values or move a value from one end of the state space to the other. The following result shows, that obtaining this *gap normal form* can be achieved with polynomial overhead as an initialization step of a BSS_K machine.

Proposition 1. *The initial state space* $(\dots, 0, 1, \dots, 1, 0.x_1, \dots, x_n, 0, \dots)$ *can be converted into* $(\dots, 0.\, x_1, 1, x_2, 1 \dots, x_n, 1, 0 \dots)$ *with quadratic overhead. Similarly, the reverse direction can be achieved with quadratic overhead.*

Remark 2. Using the gap normal form allows easy simulation of branching nodes using $x_i = x_j$ or $x_i = c$ as branching condition.

Analogously to classical complexity theory, we can argue about the complexity of a language over a semiring via the computation time of a BSS_K machine deciding it.

Definition 4. Let M be a BSS_K machine. The *input-output function* f_M of M is the partial function that maps each input $\bar{x} \in K^*$ for which the machine halts to the corresponding output of M. On any input $\bar{x} \in K^*$, the *computation time* of M on input $\bar{x}$ is the number of steps that M takes before halting. Likewise the *space* of M is the number of coordinates of the state space visited by M before halting.

The complexity class P_K (resp., PSPACE_K) is then defined as the set of all subsets of K^* that are decided by some machine M running in polynomial time (requiring polynomial space).

Non-deterministic decisions are realized using the same idea of verification as in the classical model of Turing machines. In our case, additional K-values are guessed and added after the input to the BSS_K machine by modifying the input mapping.

Definition 5. A language $L \subseteq K^*$ is *decided non-deterministically* by a BSS_K machine M, if $\bar{x} \in L$ if and

only if there exists an $\bar{x}' \in K^*$ such that $f_M(\bar{x}, \bar{x}') \neq 0$. We call $\bar{x}'$ the *guess*. Furthermore, the machine M uses a modified input mapping $g'_I \colon \mathcal{I} \to \mathcal{S}$, which places $((x_1, \dots, x_n), (x'_1, \dots, x'_m))$ in the state

$$(\dots, 0, \underbrace{1, \dots, 1}_{m}, 0, \underbrace{1, \dots, 1}_{n}, \\ 0.x_1, \dots, x_n, 0, x'_1, \dots, x'_m, 0, \dots) \in \mathcal{S}.$$

Based on this definition, we are able to define a semiring analogue to the class NP via the model of non-deterministic BSS_K machines. Note that the run time of non-deterministic BSS_K machines is determined in regard to the size of the first parameter of the input and not dependent on the guess.

Definition 6. The class NP_K consists of all languages $L \subseteq K^*$ for which there exists a BSS_K machine M that runs in polynomial time such that M decides L non-deterministically.

Similarly, we define the notion of polynomial time many-to-one reductions with respect to functions computable in polynomial time by BSS_K machines.

Definition 7 ($\leq_{\mathrm{m}}^{\mathsf{P}_K}$)**.** Let $P, Q \subseteq K^*$. We then write $P \leq_{\mathrm{m}}^{\mathsf{P}_K} Q$ if there exists a function $f \colon K^* \to K^*$ such that f is computable in polynomial time by a BSS_K machine and for all $\bar{x} \in K^*$ it holds that $x \in P$ if and only if $f(\bar{x}) \in Q$.

In line with the previous definitions, we now define the analogue of NP-completeness over a semiring K.

Definition 8. A language $L \subseteq K^*$ is NP_K*-complete* if $L \in \mathsf{NP}_K$ (membership in NP_K) and $L' \leq_{\mathrm{m}}^{\mathsf{P}_K} L$, for all languages $L' \in \mathsf{NP}_K$ (hardness for NP_K).

3 Propositional Logic

We first introduce $\mathrm{PL}_K^=$, the propositional logic over semirings. Let Prop be a countably infinite set of propositions. An assignment s is a mapping $s \colon \{\, p, \neg p \mid p \in \mathrm{Prop} \,\} \to K$ that associates each (negated) atomic proposition with a semiring value.

Definition 9. Let $K = (K, +, \cdot, 0, 1)$ be a semiring and Prop a set of propositional symbols. The *syntax of propositional logic over* K *and* Prop is defined as follows:

$$\phi ::= p \mid \neg p \mid c \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid \phi = \phi,$$

where $p \in \mathrm{Prop}$, $c \in K$. For ordered K we additionally have $\phi ::= \phi \leq \phi$.

Let $s \colon \{\, p, \neg p \mid p \in \mathrm{Prop} \,\} \to K$ be an assignment. The *semantics of a* $\mathrm{PL}_K^=$*-formula* θ *under an assignment* s is denoted by $[\![\theta]\!]_s$ and defined as follows:

$$[\![p]\!]_s = s(p), \qquad [\![\neg p]\!]_s = s(\neg p),$$
$$[\![c]\!]_s = c,$$
$$[\![\phi \wedge \psi]\!]_s = [\![\phi]\!]_s \cdot [\![\psi]\!]_s, \quad [\![\phi \vee \psi]\!]_s = [\![\phi]\!]_s + [\![\psi]\!]_s,$$
$$[\![\phi \circ \psi]\!]_s = \begin{cases} 1 & \text{if } [\![\phi]\!]_s \circ [\![\psi]\!]_s, \\ 0 & \text{otherwise,} \end{cases} \quad \text{where } \circ \in \{=, \leq\}.$$

This definition only makes use of atomic negation, which is a standard approach in semiring semantics (Green, Karvounarakis, and Tannen 2007). In this framework, $s(p)$ and $s(\neg p)$ are independent values representing the "evidence for" and "evidence against" a proposition. This is crucial for inconsistent or fuzzy information processing, where a database might contain reasons for both a fact and its negation.

Other methods to deal with negation in the setting of semiring semantics include *monus semirings* and *negation via implication*, each with their own advantages and disadvantages (Grädel and Tannen 2025, Section 8).

We will make use of shortcut notations that allow us to use any of the following comparison operators in $\mathrm{PL}_K^=$ formulae: $\phi \neq \psi \coloneqq (\phi = \psi) = 0, \phi \not\leq \psi \coloneqq (\phi \leq \psi) \leq 0$. We define this (as well as first- and second-order) logic using formula value comparisons $\phi = \psi$ as this gives us the ability to show hardness results such as Theorem 1 and Theorem 3. These constructs were originally introduced into first-order logic to enable the formalization of quantitative constraints, such as probabilistic independence, by the means of FO-formulae (Barlag et al. 2023). The upper bounds still hold true for the logics without the comparison atom.

Next, we define the satisfiability problem $\mathrm{SAT}_K^=$ for a semiring K in the context of this logic. Since there is not exactly one value denoting truth in general, we define satisfaction with respect to any non-zero value.

Definition 10. The *satisfiability problem* for $\mathrm{PL}_K^=$ is defined as

$$\mathrm{SAT}_K^= \coloneqq \{\, \varphi \in \mathrm{PL}_K^= \mid \text{ there exists an assignment } s \text{ such that } [\![\varphi]\!]_s \neq 0 \,\}.$$

Example 1. Consider the semiring $\mathbb{N}$ of natural numbers. The formula $\phi = ((p \vee q) \leq (p \wedge q)) \wedge 3 \leq p \wedge 3 \leq q$ is satisfied by the assignment $s(p) = s(q) = 3$, with $[\![\phi]\!]_s = ((3+3) \leq (3 \cdot 3)) \cdot (3 \leq 3) \cdot (3 \leq 3) = 1$. As a result, we have that $\phi \in \mathrm{SAT}_{\mathbb{N}}^=$. On the other hand, the slightly modified formula $\psi = ((p \wedge q) \leq (p \vee q)) \wedge 3 \leq p \wedge 3 \leq q$ is not satisfied by any assignment, because either $p < 3$, $q < 3$ or $p \cdot q > p + q$. Therefore, we get that $[\![\psi]\!]_s = 0$ for all assignments s and $\psi \notin \mathrm{SAT}_{\mathbb{N}}^=$.

Similar to the classical setting, we can now prove a version of Cook's theorem over semirings. That is, we show that the satisfiability problem of $\mathrm{PL}_K^=$ is also complete for the class of non-deterministic polynomial time in the semiring setting.

But first, we briefly discuss encodings of assignments and formulae. An assignment can be encoded as elements of K^*. Similarly, we can encode formulae of $\mathrm{PL}_K^=$ as elements of K^*. For these encodings, we follow the convention that elements of the semiring K are encoded as they are, i. e., each $x \in K$ is encoded by x itself, and other information is encoded using binary encodings with the semiring elements 0 and 1. Then, for instance, symbols for connectives are encoded as binary strings, whereas constants corresponding to elements of K are encoded as their interpretations in K.

Lemma 1. $\mathrm{SAT}_K^= \in \mathsf{NP}_K$.

Proof. Let M be a non-deterministic BSS_K machine with input $\langle \varphi \rangle$, the encoding of a $\mathrm{PL}_K^=$ formula, that guesses $\langle s \rangle$, the encoding of an assignment for φ. This machine M then computes $f_M(\langle \varphi \rangle, \langle s \rangle) = [\![\varphi]\!]_s$ in polynomial time, bounded by the size of the encoding $|\langle \varphi \rangle|$ of the formula φ.

Now, we have that $\varphi \in \mathrm{SAT}^{=}_{K}$ if and only if there exists an assignment s such that $[\![\varphi]\!]_s \neq 0$ if and only if there is a guess $\langle s \rangle$ such that $f_M(\langle\varphi\rangle, \langle s\rangle) \neq 0$. □

Theorem 1. *$\mathrm{SAT}^{=}_{K}$ is NP_K-complete.*

Proof. Since Lemma 1 gives us $\mathrm{SAT}^{=}_{K} \in \mathsf{NP}_K$, we only need to show hardness. We construct a formula φ that describes the computation of a non-deterministic BSS_K machine M on some arbitrary input x, such that φ is satisfiable if and only if M accepts x. To do so, we define propositions that describe the content of the tape and the current computation node of M at a given time t:

- $v_{t,p} \in K$ describes the content of the tape at time t and position p
- $q_{t,s} = \begin{cases} 1, & \text{if } M \text{ is at node } s \text{ at time } t \\ 0, & \text{else.} \end{cases}$

In contrast to the original proof for the Boolean setting, the propositions $v_{t,p}$ are values from the semiring K as we are adapting it to BSS_K machines. We then use these to build φ as a conjunction of subformulae that each describe the changes in the tape and nodes in one time step t to $t+1$ for the runtime of the machine M. Together with some subformulae that ensure a correct initialization, termination and no invalid transitions, this makes up the formula φ as desired. The full proof can be found in the earlier arXiv version of the paper (Barlag et al. 2025b). □

Remark 3. Notice that we only need formula value comparisons with constants to show hardness for NP_K.

4 First-Order Logic

In this section, we define $\mathrm{FO}^{=}_{K}$, the first-order logic over semirings, and consider some of its basic properties, as well as its model-checking problem.

4.1 Definition and Basic Properties of $\mathrm{FO}^{=}_{K}$

We begin by defining the syntax and semiring semantics for first-order logic with constants and formula value comparison atoms (cf. (Grädel and Tannen 2017; Barlag et al. 2023)).

Definition 11. The *syntax for first-order logic over a semiring K* (of vocabulary τ), is defined as follows:

$$\phi ::= x = y \mid x \neq y \mid c \mid R(\bar{x}) \mid \neg R(\bar{x}) \mid \phi = \phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid \exists x \phi \mid \forall x \phi,$$

where $x, y \in \mathrm{Var}$, $c \in K$, $R \in \tau$ and $\bar{x}$ is a tuple of variables so that $\mathrm{ar}(R) = |\bar{x}|$. For ordered semirings K, we additionally have that $\phi \leq \phi$.

Note that a formula of $\mathrm{FO}^{=}_{K}$ that does not contain any $c \in K$ or subformulae of form $\phi = \psi$ or $\phi \leq \psi$ is a first-order (FO) formula in the usual sense. The definition of a set of *free variables* $\mathrm{FV}(\phi)$ of a formula $\phi \in \mathrm{FO}^{=}_{K}$ extends from FO in the obvious way: $\mathrm{FV}(c) = \emptyset$ and $\mathrm{FV}(\phi \circ \psi) = \mathrm{FV}(\phi) \cup \mathrm{FV}(\psi)$, for $\circ \in \{=, \leq\}$. If ϕ contains no free variables, i. e., $\mathrm{FV}(\phi) = \emptyset$, it is called a *sentence*.

Let A be a finite set (i. e., the model domain), $\tau = \{R_1, \ldots, R_n\}$ a finite vocabulary, and K a semiring. Define

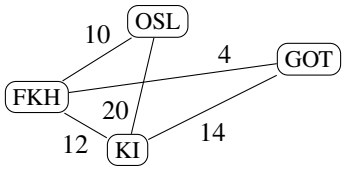

Figure 1: Ferry network as a K-interpretation.

$\mathrm{Lit}_{A,\tau}$ as the set of *literals* over A, i. e., the set of facts and negated facts $\mathrm{Lit}_{A,\tau} = \{ R(\bar{a}), \neg R(\bar{a}) \mid \bar{a} \in A^{\mathrm{ar}(R)}, R \in \tau \}$. A *$K$-interpretation* π is a mapping $\pi \colon \mathrm{Lit}_{A,\tau} \to K$. A K-interpretation π is *ordered*, if $A = \{a_0, a_1, \ldots, a_{|A|-1}\}$ and τ contains a binary relation $R_<$ such that

$$\pi(R_<(a_i, a_j)) = \begin{cases} 1, & \text{if } i < j, \\ 0, & \text{otherwise.} \end{cases}$$

An *assignment* s is a mapping $s \colon \mathrm{Var} \to A$ that associates each first-order variable with an element of A.

Definition 12. Let $K = (K, +, \cdot, 0, 1)$ be a semiring, π be a K-interpretation, and s be an assignment. The *semantics of a $\mathrm{FO}^{=}_{K}$-formula θ under π and s*, is denoted by $[\![\theta]\!]_{\pi,s}$ and defined as follows:

$$[\![x \star y]\!]_{\pi,s} = \begin{cases} 1 \text{ if } s(x) \star s(y), \\ 0 \text{ otherwise,} \end{cases} \quad \text{where } \star \in \{=, \neq\},$$

$$[\![c]\!]_{\pi,s} = c$$

$$[\![R(\bar{x})]\!]_{\pi,s} = \pi(R(s(\bar{x}))), \quad [\![\neg R(\bar{x})]\!]_{\pi,s} = \pi(\neg R(s(\bar{x}))),$$

$$[\![\phi \wedge \psi]\!]_{\pi,s} = [\![\phi]\!]_{\pi,s} \cdot [\![\psi]\!]_{\pi,s}, \quad [\![\exists x \phi]\!]_{\pi,s} = \sum_{a \in A} [\![\phi]\!]_{\pi, s(a/x)},$$

$$[\![\phi \vee \psi]\!]_{\pi,s} = [\![\phi]\!]_{\pi,s} + [\![\psi]\!]_{\pi,s}, \quad [\![\forall x \phi]\!]_{\pi,s} = \prod_{a \in A} [\![\phi]\!]_{\pi, s(a/x)},$$

$$[\![\phi \circ \psi]\!]_{\pi,s} = \begin{cases} 1 \text{ if } [\![\phi]\!]_{\pi,s} \circ [\![\psi]\!]_{\pi,s}, \\ 0 \text{ otherwise,} \end{cases} \quad \text{where } \circ \in \{=, \leq\}.$$

If ϕ is a *sentence*, we write $[\![\phi]\!]_\pi$ for $[\![\phi]\!]_{\pi,\emptyset}$.

We can again define $\neq$ and $\not\leq$ for formulae, in the same way as was done for $\mathrm{PL}^{=}_{K}$. Sometimes, K-interpretations π are called *model-defining* (Grädel and Tannen 2017; Grädel and Tannen 2025), which is the case if, for all $R(\bar{a})$, we have that $\pi(R(\bar{a})) = 0$ if and only if $\pi(\neg R(\bar{a})) \neq 0$. Sentences ϕ and ψ of $\mathrm{FO}^{=}_{K}$ are *K-equivalent*, written as $\phi \equiv_K \psi$, if $[\![\phi]\!]_\pi = [\![\psi]\!]_\pi$ is true for all model-defining K-interpretations.

Example 2. Suppose A consists of the city codes OSL (Oslo), GOT (Göteborg), KI (Kiel), FKH (Frederikshavn), and τ of one edge relation E. Consider the tropical semiring $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ with its natural order (defined for semirings K as: $a \leq b$ if and only if $\exists c : a + c = b$). In particular, since addition is interpreted as $\min$, the natural order inverts the usual order of the reals. An example $\mathbb{T}$-interpretation $\pi \colon Lit_{A,\tau} \to \mathbb{T}$ is obtained by assigning $E(x, y)$, for each pair of cities x and y, a number representing the duration of a direct ferry ride between them, as in Figure 1; if no direct ferry connection exists between two cities x and y, $E(x, y)$ is assigned ∞. Furthermore each

negated fact $\neg E(x, y)$ can be assigned a number so that π is model-defining. Then, we can express in $\mathrm{FO}^{=}_{K}$ that the duration of a direct ferry trip between any two cities x and y is shorter than the sum of durations for ferries from x to z and z to y, for any city z:

$$\phi \coloneqq \forall xyz((E(x, z) \wedge E(z, y)) \leq E(x, y)). \qquad (1)$$

Note that $[\![\phi]\!]_\pi = \infty$, that is, ϕ evaluates to the identity element of addition under π (i.e., ϕ is "false"), since OSL and GOT are missing a direct connection. Adding a connection between these two cities, labeled by 8, would entail $[\![\phi]\!]_\pi = 0$ (i.e., ϕ is "true"). Furthermore, in the naturally ordered Boolean semiring, (1) simply expresses that the graph is transitively closed, i.e., "$\leq$" collapses to the ordinary Boolean implication "$\rightarrow$".

The previous example demonstrates that $\mathrm{FO}^{=}_{K}$ provides a principled generalization of first-order logic to the semiring context. In addition to interpreting disjunction as addition and conjunction as multiplication, implication can be conceived via the (natural) order of the semiring. As a consequence, the same sentence can express either transitivity in the Boolean setting or the triangle inequality in the tropical setting, depending on the choice of the semiring. From this perspective, classical first-order logic arises as the special case induced by the Boolean semiring, where the natural order coincides with ordinary implication.

For logics $\mathcal{L}$ and $\mathcal{L}'$, we write $\mathcal{L} \leq_K \mathcal{L}'$ if for each sentence ϕ from $\mathcal{L}$ there is a sentence ψ from $\mathcal{L}'$ such that $\phi \equiv_K \psi$. If $\mathcal{L} \leq_K \mathcal{L}'$ and $\mathcal{L}' \leq_K \mathcal{L}$, we write $\mathcal{L} \equiv_K \mathcal{L}'$, and say that $\mathcal{L}$ and $\mathcal{L}'$ are *equally expressive under* K. As usual, we write $\mathcal{L} <_K \mathcal{L}'$ if $\mathcal{L} \leq_K \mathcal{L}'$ and $\mathcal{L}' \not\leq_K \mathcal{L}$. We now consider some results that demonstrate the expressivity of the logics in the special case of the Boolean semiring $\mathbb{B} = (\mathbb{B}, \vee, \wedge, 0, 1)$. Let $\mathcal{A}$ be a structure of vocabulary τ. The canonical truth interpretation $\pi_{\mathcal{A}}$, is the $\mathbb{B}$-interpretation $\pi\colon \mathrm{Lit}_{A,\tau} \to \mathbb{B}$ defined for each $R(\bar{a})$ and $\neg R(\bar{a})$ as follows: $\pi(R(\bar{a})) = 1$ if $\bar{a} \in R^{\mathcal{A}}$, $\pi(R(\bar{a})) = 0$ if $\bar{a} \notin R^{\mathcal{A}}$, and $\pi(\neg R(\bar{a})) = 1 - \pi(R(\bar{a}))$. The following proposition shows that for FO-formulae, the canonical truth interpretation $\pi_{\mathcal{A}}$ under an assignment s corresponds to the usual first-order formula evaluation under s in the structure $\mathcal{A}$. It can be proven by induction on α.

Proposition 2 (Grädel and Tannen 2017)**.** *Let α be an* FO*-formula, and $\mathcal{A}$ a structure. Then $[\![\alpha]\!]_{\pi_{\mathcal{A}},s} = 1$ if and only if $\mathcal{A} \models_s \alpha$.*

Let $\xi_K\colon K \to \mathbb{B}$ be the characteristic mapping such that $\xi_K(a) = 0$ if a is the zero element of K and $\xi_K(a) = 1$ otherwise. Collapsing the interpretations of FO-formulae to the Boolean semiring by taking the composition of the characteristic mapping ξ_K and a K-interpretation π preserves the 0-valued interpretations.

Proposition 3 (Barlag et al. 2023)**.** *Let be $\pi\colon \mathrm{Lit}_{A,\tau} \to K$ an interpretation. Then for all $\alpha \in \mathrm{FO}$, $[\![\alpha]\!]_{\xi_K \circ \pi} = 1$ if and only if $[\![\alpha]\!]_\pi \neq 0$.*

Note that in case of the Boolean semiring $\mathbb{B}$, the constants 0 and 1 can be expressed with the FO-formulae $x \neq x$ and $x = x$, respectively. This means that adding these constants does not affect the result of (Barlag et al. 2023) that states that having access to formula value comparison atoms does not increase expressivity in the sense of the following proposition.

Proposition 4 (Barlag et al. 2023)**.** $\mathrm{FO} \equiv_{\mathbb{B}} \mathrm{FO}^{=}_{\mathbb{B}}$.

The above proposition does not hold if the semiring K has at least three elements. Note that constant formulae are not needed for the result: there are constant-free $\mathrm{FO}^{=}_{K}$-formulae that do not have K-equivalent FO-formulae.

Proposition 5. *Let K be a semiring such that $|K| \geq 3$. Then we have that* $\mathrm{FO} <_K \mathrm{FO}^{=}_{K}$.

Proof. Notice first that if K is ordered and $|K| \geq 3$, there are two non-zero elements $a_1, a_2 \in K$ such that $a_1 \not\leq a_2$. Either all non-zero elements of K are incomparable (and hence the existence of the two elements follows trivially) or there are two different non-zero elements $b_1, b_2 \in K$ such that $b_1 \leq b_2$. In the latter case, by antisymmetry of the order, we have $b_2 \not\leq b_1$. Consider the formula $\psi \coloneqq \forall x \forall y(P(x) \circ P(y))$, where $\circ \in \{=, \leq\}$ We claim that ψ cannot be translated to FO. Suppose for a contradiction that we find $\psi' \in \mathrm{FO}$ such that $\psi \equiv_K \psi'$. Let $A = \{1, 2\}$, and let $\tau = \{P\}$ for a unary relation symbol P. Consider two interpretations $\pi\colon \mathrm{Lit}_{A,\tau} \to K$ and $\pi'\colon \mathrm{Lit}_{A,\tau} \to K$ such that $\pi(P(i)) = a_i$ and $\pi'(P(i)) = 1$ for all $i \in A$, where a_1, a_2 are two different non-zero elements (such that $a_1 \not\leq a_2$). Now $[\![\psi]\!]_\pi = 0$, because $[\![P(1) \circ P(2)]\!]_\pi = 0$. In contrast, since $[\![P(i) \circ P(j)]\!]_{\pi'} = 1$ for all $i, j \in \{1, 2\}$, we obtain $[\![\psi]\!]_{\pi'} = 1$. Consequently, our hypothesis entails $[\![\psi']\!]_\pi = 0$ and $[\![\psi']\!]_{\pi'} = 1$, so by Proposition 3, $[\![\psi']\!]_{\xi_K \circ \pi} = 0$ and $[\![\psi']\!]_{\xi_K \circ \pi'} = 1$. But since $\xi_K \circ \pi = \xi_K \circ \pi'$, this is impossible. Thus the claim follows. □

4.2 Model-checking of FO formulae

In every logic, model-checking is a central problem of interest. In the following, we define the model-checking problem in the setting of combined complexity.

Problem:	$\mathrm{FO}^{=}_{K}$-MC — the model-checking problem for $\mathrm{FO}^{=}_{K}$ formulae
Input:	$\mathrm{FO}^{=}_{K}$ formula φ, K-interpretation π, assignment s
Question:	$[\![\varphi]\!]_{\pi,s} \neq 0$?

Given a formula φ, we write $\mathrm{FO}^{=}_{K}$-MC(φ) to denote the fragment of $\mathrm{FO}^{=}_{K}$-MC restricted to the fixed formula φ.

The encoding of the input formulae and K-interpretations can be done analogously to the formulae and assignments in the case of $\mathrm{PL}^{=}_{K}$. Note that in the case of $\mathrm{FO}^{=}_{K}$, assignments s contain no semiring values and can thus be encoded using 0 and 1 in the usual way. Lastly, notice that while the universe A is absent from the inputs to $\mathrm{FO}^{=}_{K}$-MC, it is implicitly given by the K-interpretation π.

Theorem 2. *Fix a semiring K. Given a $\mathrm{FO}^{=}_{K}$ formula φ, a K interpretation π, and an assignment s. The value $[\![\varphi]\!]_{\pi,s}$ can be computed by a BSS_K machine in time $O(n^2 \cdot |\varphi| \cdot |\pi|^{|\varphi|})$ and in space $O(\mathrm{poly}(n))$, where $n = |\varphi| + |\pi| + |s|$ is the sum of the encoding lengths.*

The algorithm and runtime analysis can be found in the earlier arXiv version of the paper (Barlag et al. 2025a).

Corollary 1. *Every $f \in \mathrm{FO}^{=}_{K}$ can be computed in polynomial space.*

We immediately get model-checking results from Theorem 2 by merely checking whether the computed value $[\![\varphi]\!]_{\pi,s}$ is not 0.

Corollary 2. *For any $\phi \in \mathrm{FO}$, $\mathrm{FO}^{=}_{K}$-$\mathrm{MC}(\varphi) \in \mathsf{P}_K$.*

Corollary 3. $\mathrm{FO}^{=}_{K}$-$\mathrm{MC} \in \mathsf{PSPACE}_K$.

5 Existential Second-Order Logic

We define existential second-order quantification such that it corresponds to the existence of a K-relation in the Boolean sense, i. e., a formula with second-order quantification can only have values 0 or 1. Another natural option would be to define existential second-order quantification analogously to existential first-order quantification, by using a sum over all possible K-interpretations of the quantified relation. Note that in our setting, we are mainly interested in infinite semirings, since the operation table of a finite semiring can be handled with a standard Turing machine. This means that if the semantics of second-order quantification is defined by using sums, we would need to consider infinite (even uncountable) sums as described in more detail by (Brinke et al. 2024). Also note that second-order existential quantification over infinite semirings is problematic because using the standard approach would require infinite sums to be definable, which is not always the case.

Definition 13 ($\mathrm{ESO}^{=}_{K}$)**.** The *syntax for existential second-order logic* (of vocabulary τ) over a (ordered) semiring K, is defined as follows

$$\phi ::= \alpha \mid \exists R\phi,$$

where $\alpha \in \mathrm{FO}^{=}_{K}$ and R is a relation symbol not in τ. We say that a K-interpretation $\pi' \colon \mathrm{Lit}_{A,\tau\cup\{R\}} \to K$ *extends* π *with* R if $\pi'(\ell) = \pi(\ell)$ for all $\ell \in \mathrm{Lit}_{A,\tau}$. The *semantics of existential second-order quantification* is then defined as follows.

$$[\![\exists R\phi]\!]_{\pi,s} := \begin{cases} 1, & \text{if there is } \pi' \text{ that extends } \pi \text{ with} \\ & R \text{ such that } [\![\phi]\!]_{\pi',s} \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

This definition enforces that all quantified relation symbols are in the beginning of the formula. Allowing quantified relations to appear anywhere within a formula, especially inside comparisons, would allow expressive power beyond existential second-order logic. This is the case, because it would allow that negated existential second-order quantifier could be constructed, e. g., $\neg\exists P\psi$ could be simulated by $(\exists P\psi) = 0$.

Example 3 (Euclidian Distance Matrix recognition). Given $k \in \mathbb{N}$, a k-dimensional Euclidian Distance Matrix (EDM) D is an n-by-n matrix such that there exists $k \in [n]$ and $p_1, \ldots, p_n \in \mathbb{R}^k$ having

$$D_{ij} = \|p_i - p_j\|_2^2,$$

for all $i, j \in [n]$. This means, in other words, that n points can be embedded into the k-dimensional Euclidean space in such a way that the distances between all pairs of vertices agree with D. Consider a formula of $\mathrm{ESO}^{=}_{K}$ defined as:

$$\phi := \exists\vec{R}\vec{S}\forall xy\Big(\bigwedge_{i=1}^{k}(R_i(x) \vee S_i(x,y) = R_i(y)) \wedge (S_1(x,y)^2 \vee \cdots \vee S_k(x,y)^2 = c_{ij})\Big),$$

where θ^2 is a shorthand for $\theta \wedge \theta$. Let $A = [n]$ and let $\tau = \{c_{ij} \mid i, j \in\}$, where c_{ij} are nullary relation symbols (i.e., constant symbols). Consider a $\mathbb{R}_{\geq 0}$-interpretation $\pi \colon \mathrm{Lit}_{A,\tau} \to \mathbb{R}_{\geq 0}$ such that $[\![c_{ij}()]\!]_\pi = D_{ij}$. Then, $[\![c_{ij}()]\!]_\pi \neq 0$ if and only if D is a k-dimensional EDM.

We now present a version of Fagin's theorem in the context of semiring semantics, which is used to characterise the class NP_K via existential second-order logic over K. The proof follows the classical structure, but has been adapted to work with our model of BSS_K machines. From the logical perspective, we consider ordered K-interpretations in which a single unary K-relation I is used to encode an input in K^*. Formulae and K-interpretations are encoded as described in Section 4.2.

Lemma 2. $\mathrm{ESO}^{=}_{K} \subseteq \mathsf{NP}_K$.

Proof. Let $\phi := \exists R_1 \ldots \exists R_k \psi$ be an existential second-order formula with quantified relation symbols $R_1, \ldots, R_k$. Per definition, we have that $[\![\phi]\!]_\pi \neq 0$ for a K-interpretation π if there exists a K-interpretation π' for the symbols $R_1, \ldots, R_k$ such that $[\![\psi]\!]_{\pi\cup\pi'} \neq 0$. There exists a BSS_K machine M that deterministically evaluates ψ in polynomial time given the encoding of a K-interpretation $\langle\pi\cup\pi'\rangle$ as input (see Corollary 2). Thus, there is a non-deterministic BSS_K machine M' that, with input $\langle\pi\rangle$, guesses $\langle\pi'\rangle$, and accepts if and only if $[\![\phi]\!]_\pi \neq 0$. Furthermore, it holds that $[\![\Phi]\!]_\pi \neq 0$ if only if $[\![\Psi]\!]_{\pi\cup\pi'} \neq 0$ if and only if $f_{M'}(\langle\pi\rangle, \langle\pi'\rangle) \neq 0$. □

Theorem 3. $\mathrm{ESO}^{=}_{K} = \mathsf{NP}_K$.

Proof sketch. Since Lemma 2 gives us $\mathrm{ESO}^{=}_{K} \subseteq \mathsf{NP}_K$, only $\mathsf{NP}_K \subseteq \mathrm{ESO}^{=}_{K}$, is left to prove. The proof is similar to the original proof in the Boolean setting, but is altered to use BSS_K machines and therefore uses values that are elements of the semiring K. We construct a formula $\Phi \in \mathrm{ESO}^{=}_{K}$ that describes for a non-deterministic BSS_K machine M on some arbitrary input x, that there exists a computation such that M accepts. To do so we define relations $\bar{t} = (t_1, \ldots, t_z) \in A^z$ and $\bar{p} = (d, p_1, \ldots, p_z) \in A^{z+1}$, where $A = \{a_0, \ldots, a_{n-1}\}$ is our domain, to identify the time step and position of M.

We then use these to build Φ as an existentially quantified conjunction of subformulae that each describe the changes in the tape and nodes in one time step t to $t + 1$ for the runtime of the machine M. Together with subformulae that ensure a correct initialization, termination and no invalid transitions, this makes up the formula Φ as desired. The full proof can be found in the earlier arXiv version of the paper (Barlag et al. 2025b). □

6 Non-arithmetic Reductions and the Existential Theory of K

In this section, we define novel non-arithmetic machines and use them to show that the existential first-order theory of a semiring K is complete for the Boolean fragment of the complexity class NP_K. We also generalize this to BSS_K machines that operate on subsets of K. For this, we turn our attention to machines M that correspond to polynomial-time computable input-output functions of the forms $f_M\colon X^* \to K^*$ and $f_M\colon X^* \to X^*$ for some $X \subseteq K$. However, without loss of generality, we may assume that these machines run in polynomial time for all inputs in the set K^*. This simplifying observation is an application of using a binary counter to limit the number of computation steps before always halting. The missing proofs of this section can be found in the earlier arXiv version of the paper (Barlag et al. 2025b).

Lemma 3. *Let $X \subseteq K$ and let $f\colon X^* \to K^*$ be a function computed by a BSS_K machine M that runs in polynomial time on inputs in the set X^*. Then there is a BSS_K machine M' such that $f_{M'} \restriction X^* = f$ and that M' runs in polynomial time on all inputs in K^*. Furthermore, the machine M' is limited to only use the machine constants of M together with 0 and 1.*

Definition 14. Let $\{0,1\} \subseteq X \subseteq K$. The class $\mathsf{NP}_K[X]$ consists of all languages $L \subseteq X^*$ for which there exists some BSS_K machine M that runs in polynomial time and decides L non-deterministically among inputs in X^* and uses only machine constants that are in the set X. That is, for every input $\bar{x} \in X^*$ it holds that $\bar{x} \in L$, if and only if for some guess $\bar{x}' \in K^*$, $f_M(\bar{x}, \bar{x}') \neq 0$.

Definition 15. For languages $L \subseteq K^*$ and $L' \subseteq X^*$, we write $L' \leq_{\mathrm{m}}^{\mathsf{P}_K[X]} L$ if there is a function $f\colon X^* \to K^*$ that is computable in polynomial time by a BSS_K machine using only machine constants in the set X such that for all $\bar{x} \in X^*$ it holds that $\bar{x} \in L'$ if and only if $f(\bar{x}) \in L$.

In particular, $\mathsf{NP}_K[K] = \mathsf{NP}_K$, and for all languages $L, L' \subseteq K^*$ the conditions $L' \leq_{\mathrm{m}}^{\mathsf{P}_K} L$ and $L' \leq_{\mathrm{m}}^{\mathsf{P}_K[K]} L$ are equivalent. We do not assume that the subsets $X \subseteq K$ we consider are closed under the arithmetic operations of the semiring K, but instead we focus on reductions based on BSS_K machines that use arithmetic operations only for copying arbitrary elements of the semiring without changing them, and for assigning constant values, thus resembling Turing machines.

Definition 16. A BSS_K machine M is said to be *non-arithmetic*, if for every input $\bar{x} \in K^*$ it holds that whenever an arithmetic computation node is reached during a computation, this computation node corresponds to either a constant assignment or an arithmetic operation $a \star b$ for $\star \in \{+, \cdot\}$ so that at least one of the operands a and b is the neutral element of the operation $\star$ in the semiring K. Furthermore, a computable function f is *non-arithmetic*, if it is computed by some non-arithmetic machine. For $\{0,1\} \subseteq X \subseteq K$ and $P, Q \subseteq X^*$, a reduction $Q \leq_{\mathrm{m}}^{\mathsf{P}_K[X]} P$ is called *non-arithmetic*, if the computable function corresponding to it is non-arithmetic.

Lemma 4. *Let $X \subseteq K$ be a finite subset, $\{0,1\} \subseteq X$, and let $P, Q \subseteq X^*$. Then there is a non-arithmetic reduction $Q \leq_{\mathrm{m}}^{\mathsf{P}_K[X]} P$, if and only if we have $Q \leq_{\mathrm{m}}^{\mathsf{P}} P$, that is, there is a Turing machine with input alphabet X computing a reduction from Q to P in polynomial time.*

We give two definitions for completeness based on non-arithmetic reductions. The first one is a non-arithmetic version of $\mathsf{NP}_K[X]$-completeness, and the other one is an even stronger form of non-arithmetic NP_K-completeness.

Definition 17. Let $\{0,1\} \subseteq X \subseteq K$ and $L \subseteq X^*$. We say that L is $\mathsf{NP}_K[X]$*-complete under non-arithmetic reductions*, if $L \in \mathsf{NP}_K[X]$ and if for every $L' \in \mathsf{NP}_K[X]$ there is a non-arithmetic reduction $L' \leq_{\mathrm{m}}^{\mathsf{P}_K} L$.

Definition 18. A language $L \subseteq K^*$ is NP_K*-complete under non-arithmetic reductions respecting machine constants*, if the membership $L \in \mathsf{NP}_K$ is witnessed by a polynomial-time BSS_K machine that only uses 0 and 1 as machine constants, and if for every $L' \in \mathsf{NP}_K$ and for every polynomial-time BSS_K machine M the following holds: if M decides L' non-deterministically, then there is a polynomial-time non-arithmetic BSS_K machine M' that is limited to only use the machine constants of M together with 0 and 1 in such a manner that for every $\bar{x} \in K^*$ it is true that $\bar{x} \in L'$ if and only if $f_{M'}(\bar{x}) \in L$.

Lemma 5. *Let $\{0,1\} \subseteq X \subseteq K$, and let $P \subseteq K^*$ be NP_K-complete under non-arithmetic reductions that respect machine constants. Then the problem $P \cap X^*$ is $\mathsf{NP}_K[X]$-complete under non-arithmetic reductions.*

For the next result, let $\mathrm{SAT}_K^{\mathrm{flat}}$ be the restriction of $\mathrm{SAT}_K^{=}$ to those propositional formulae that do not have nested formula (in)equalities. For this, we consider the shorthand notations for the negated formula inequalities $\neq$ and $\not\leq$ to be logical symbols in their own right.

Theorem 4. *The problems $\mathrm{SAT}_K^{=}$ and $\mathrm{SAT}_K^{\mathrm{flat}}$ are NP_K-complete under non-arithmetic reductions that respect machine constants.*

Proof. Although not explicitly stated, the claims in the statement are demonstrated in the proofs of Lemma 1 and Theorem 1. For Lemma 1, machine constants for elements other than 0 and 1 are not used when a guess for a satisfying assignment is verified. The proof of Theorem 1 can be modified to show that, in addition to $\mathrm{SAT}_K^{=}$ being NP_K-complete, also the modified problem $\mathrm{SAT}_K^{\mathrm{flat}}$ is NP_K-complete using non-arithmetic machines that satisfy the claimed condition on respecting machine constants. □

Next we consider the semiring K as a structure in the context of ordinary first-order logic.

Definition 19. Let $\{0,1\} \subseteq X \subseteq K$. The existential theory of K with constants in X is defined as the set of existential sentences of first-order logic (in the vocabulary $\{+, \cdot\} \cup X$) that are true in K, i. e., as the set

$$\mathsf{ETK}[X] = \{\, \phi \in \mathrm{FO} \mid K \models \phi \,\},$$

where the standard Tarskian semantics of truth is applied. Furthermore, we write ETK instead of $\mathsf{ETK}[\{0,1\}]$. Then,

$\exists K$ is defined as the complexity class that consists of all those Boolean languages that have a polynomial-time many-one reduction to ETK, i. e.,

$$\exists K \coloneqq \{\, L \subseteq \{0,1\}^* \mid L \leq^{\mathsf{P}}_{\mathrm{m}} \mathsf{ETK} \,\}.$$

Theorem 5. *The problem $\mathsf{ETK}[K]$ is NP_K-complete under non-arithmetic reductions respecting machine constants.*

Proof. The claim $\mathsf{ETK}[K] \in \mathsf{NP}_K$ can be demonstrated by guessing a satisfying assignment for the existentially quantified variables and verifying it in polynomial time. Only the values 0 and 1 are used as machine constants.

For the hardness claim, note that since the composition of two non-arithmetic reductions is a non-arithmetic reduction, by Theorem 4 it is enough to show that there is a non-arithmetic reduction $\mathrm{SAT}^{\mathrm{flat}}_K \leq^{\mathsf{P}_K}_{\mathrm{m}} \mathsf{ETK}[K]$ that does not use machine constants other than 0 and 1. For this, let ϕ be a propositional formula that is given as input. Let g be an injection that maps each negated and non-negated propositional literal of ϕ to a first-order variable. Within formula (in)equalities, the connectives $\wedge$ and $\vee$ are replaced with $\cdot$ and $+$, respectively, and each literal ℓ is replaced with $g(\ell)$. Outside the scope of formula (in)equalities, each literal ℓ is replaced with $g(\ell) \neq 0$. Constants are kept unchanged. Now, we have a first-order formula ψ without quantifiers. Then, let $\theta \coloneqq \exists x_1 \ldots \exists x_k \psi$, where $x_1, \ldots x_k$ are the newly introduced first-order variables of ψ. Using arguments based on induction, it can be shown that the formula ϕ is satisfiable if and only if $K \models \theta$. The sentence θ can be constructed with a non-arithmetic BSS_K machine in polynomial time. □

Following our conventions regarding encodings in Sections 3 and 4, it holds that $\mathsf{ETK}[X] = \mathsf{ETK}[K] \cap X^*$ whenever $\{0,1\} \subseteq X \subseteq K$. Thus, the following result follows from Lemma 5 and Theorem 5.

Corollary 4. *Let $\{0,1\} \subseteq X \subseteq K$. Then $\mathsf{ETK}[X]$ is $\mathsf{NP}_K[X]$-complete under non-arithmetic reductions.*

In the previous result, in the case that the subset X is finite, we can use Lemma 4 to obtain a result about complexity classes for ordinary Turing machines.

Corollary 5. *Let $X \subseteq K$ be a finite set so that $\{0,1\} \subseteq X$. Then $\mathsf{ETK}[X]$ is complete for $\mathsf{NP}_K[X]$ under polynomial-time reductions computed by Turing machines with input alphabet X. In particular, $\exists K = \mathsf{NP}_K[\{0,1\}]$.*

Using the notation of, e. g., (Bürgisser and Cucker 2003), our definition of $\mathsf{NP}_K[\{0,1\}]$ coincides with the class that could be denoted by $\mathrm{BP}(\mathsf{NP}^0_K)$, the Boolean part of NP^0_K. Here, NP^0_K is the restricted version of the definition of NP_K to those BSS_K machines that are only allowed to use the semiring values 0 and 1 as machine constants. Then, $\mathrm{BP}(\mathsf{NP}^0_K)$ is defined as the set $\{\, L \cap \{0,1\}^* \mid L \in \mathsf{NP}^0_K \,\}$.

7 Conclusion

In this paper, we connected semiring semantics for logic to algebraic models of computation. As the main results, we provided semiring analogues of the classical theorems of Fagin and Cook over non-deterministic BSS_K machines. Specifically, under semiring semantics with built-in comparison atoms, existential second-order logic captures NP_K, and propositional satisfiability is complete for this class. Additionally, we showed that the existential first-order theory of a semiring K is complete for the Boolean fragment of the complexity class NP_K.

From a broader perspective, our results contribute to a systematic understanding of how classical notions from descriptive complexity theory extend beyond the Boolean setting. Future directions include relaxing algebraic assumptions (e.g., non-commutative or non-positive semirings), semiring analogues of richer complexity classes, and sharper model-checking bounds parameterized by formula and semiring properties.

Acknowledgements

The authors appreciate the project support by a Finnish-German co-operation Grant: DAAD (German Exchange Service, project id 57710940), Research Council of Finland (project id 359650). The second and eighth author appreciate funding by the German Research Foundation (grant id, 511769688, ME 4279/3-1). The fifth author appreciates funding by the Magnus Ehrnrooth Foundation.

AI Declaration

Generative AI was used to check the grammatical accuracy of the text and to restructure parts of the introduction.

References

Atserias, A., and Kolaitis, P. G. 2025. Consistency of relations over monoids. *J. ACM* 72(3):18:1–18:47.

Badia, G.; Droste, M.; Noguera, C.; and Paul, E. 2024. Logical Characterizations of Weighted Complexity Classes. In Královic, R., and Kucera, A., eds., *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, Bratislava, Slovakia, August 26-30, 2024*, volume 306 of *LIPIcs*, 14:1–14:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Badia, G.; Droste, M.; Eiter, T.; Kiesel, R.; Noguera, C.; and Paul, E. 2025. Fagin's Theorem for Semiring Turing Machines. *CoRR* abs/2507.18375.

Barlag, T.; Hannula, M.; Kontinen, J.; Pardal, N.; and Virtema, J. 2023. Unified foundations of team semantics via semirings. In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, 75–85.

Barlag, T.; Fröhlich, N.; Hankala, T.; Hannula, M.; Hirvonen, M.; Holzapfel, V.; Kontinen, J.; Meier, A.; and Strieker, L. 2025a. Logic and computation through the lens of semirings.

Barlag, T.; Fröhlich, N.; Hankala, T.; Hannula, M.; Hirvonen, M.; Holzapfel, V.; Kontinen, J.; Meier, A.; and Strieker, L. 2025b. Logical approaches to non-deterministic polynomial time over semirings.

Bizière, C.; Grädel, E.; and Naaf, M. 2023. Locality theorems in semiring semantics. In Leroux, J.; Lombardy, S.;

and Peleg, D., eds., *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, Bordeaux, France, August 28 - September 1, 2023*, volume 272 of *LIPIcs*, 20:1–20:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Blum, L.; Shub, M.; and Smale, S. 1988. On a Theory of Computation over the Real Numbers; NP Completeness, Recursive Functions and Universal Machines (Extended Abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, 387–397. IEEE Computer Society.

Bourgaux, C., and Ozaki, A. 2019. Querying Attributed DL-lite Ontologies Using Provenance Semirings. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2719–2726. AAAI Press.

Bourgaux, C.; Ozaki, A.; and Peñaloza, R. 2023. Semiring provenance for lightweight description logics. *CoRR* abs/2310.16472.

Bournez, O.; Cucker, F.; de Naurois, P. J.; and Marion, J. 2006. Implicit complexity over an arbitrary structure: Quantifier alternations. *Inf. Comput.* 204(2):210–230.

Brinke, S.; Grädel, E.; Mrkonjic, L.; and Naaf, M. 2024. Semiring provenance in the infinite. In Amarilli, A., and Deutsch, A., eds., *The Provenance of Elegance in Computation - Essays Dedicated to Val Tannen, Tannen's Festschrift, University of Pennsylvania, Philadelphia, PA, USA, May 24-25, 2024*, OASIcs, 3:1–3:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Brinke, S.; Grädel, E.; and Mrkonjic, L. 2024. Ehrenfeucht-fraïssé Games in Semiring Semantics. In *CSL*, volume 288 of *LIPIcs*, 19:1–19:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Bürgisser, P., and Cucker, F. 2003. Counting complexity classes for numeric computations II: algebraic and semialgebraic sets. *CoRR* cs.CC/0312007.

Calautti, M.; Livshits, E.; Pieris, A.; and Schneider, M. 2024. The complexity of why-provenance for datalog queries. *Proc. ACM Manag. Data* 2(2):83.

Calvanese, D.; Lanti, D.; Ozaki, A.; Peñaloza, R.; and Xiao, G. 2019. Enriching ontology-based data access with provenance. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 1616–1623. ijcai.org.

Dannert, K. M.; Grädel, E.; Naaf, M.; and Tannen, V. 2021. Semiring Provenance for Fixed-point Logic. In Baier, C., and Goubault-Larrecq, J., eds., *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, Ljubljana, Slovenia (Virtual Conference), January 25-28, 2021*, volume 183 of *LIPIcs*, 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

de Naurois, P. J. 2004. *Completeness results and syntactic characterizations of complexity classes over arbitrary structures*. Ph.D. Dissertation, National Polytechnic Institute of Lorraine, Nancy, France.

Eiter, T., and Kiesel, R. 2023. Semiring Reasoning Frameworks in AI and Their Computational Complexity. *J. Artif. Intell. Res.* 77:207–293.

Eldar, I.; Carmeli, N.; and Kimelfeld, B. 2024. Direct access for answers to conjunctive queries with aggregation. In Cormode, G., and Shekelyan, M., eds., *27th International Conference on Database Theory, ICDT 2024, Paestum, Italy, March 25-28, 2024*, volume 290 of *LIPIcs*, 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Esparza, J.; Luttenberger, M.; and Schlund, M. 2015. FP-SOLVE: A Generic Solver for Fixpoint Equations Over Semirings. *Int. J. Found. Comput. Sci.* 26(7):805–826.

Fröhlich, N.; Meier, A.; Pardal, N.; and Virtema, J. 2025. A Logic-Based Framework for Database Repairs. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning*, 326–335.

Gaubert, S., and Katz, R. 2006. Reachability Problems for Products of Matrices in Semirings. *Int. J. Algebra Comput.* 16(3):603–627.

Goode, J. B. 1994. Accessible Telephone Directories. *J. Symb. Log.* 59(1):92–105.

Grädel, E., and Meer, K. 1995. Descriptive complexity theory over the real numbers. In Leighton, F. T., and Borodin, A., eds., *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, 315–324. ACM.

Grädel, E., and Mrkonjic, L. 2021. Elementary equivalence versus isomorphism in semiring semantics. In Bansal, N.; Merelli, E.; and Worrell, J., eds., *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, volume 198 of *LIPIcs*, 133:1–133:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Grädel, E., and Tannen, V. 2017. Semiring Provenance for First-order Model Checking. *CoRR* abs/1712.01980.

Grädel, E., and Tannen, V. 2025. Provenance analysis and semiring semantics for first-order logic. In Meer, K.; Rabinovich, A.; Ravve, E.; and Villaveces, A., eds., *Model Theory, Computer Science, and Graph Polynomials: Festschrift in Honor of Johann A. Makowsky*. Cham: Springer Nature Switzerland. 351–401.

Grädel, E.; Helal, H.; Naaf, M.; and Wilke, R. 2022. Zero-one laws and almost sure valuations of first-order logic in semiring semantics. In Baier, C., and Fisman, D., eds., *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, 41:1–41:12. ACM.

Green, T. J., and Tannen, V. 2017. The semiring framework for database provenance. In Sallinger, E.; den Bussche, J. V.; and Geerts, F., eds., *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, 93–99. ACM.

Green, T. J.; Karvounarakis, G.; and Tannen, V. 2007. Provenance semirings. In Libkin, L., ed., *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, 31–40. ACM.

Grohe, M.; Standke, C.; Steegmans, J.; and den Bussche, J. V. 2025. Query languages for neural networks. In Roy, S., and Kara, A., eds., *28th International Conference on Database Theory, ICDT 2025, Barcelona, Spain, March 25-28, 2025*, volume 328 of *LIPIcs*, 9:1–9:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Im, S.; Moseley, B.; Ngo, H.; and Pruhs, K. 2024. On the Convergence Rate of Linear Datalog° over Stable Semirings. In Cormode, G., and Shekelyan, M., eds., *27th International Conference on Database Theory (ICDT 2024)*, volume 290 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 11:1–11:20. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Khamis, M. A.; Ngo, H. Q.; Pichler, R.; Suciu, D.; and Wang, Y. R. 2024. Convergence of datalog over (pre-) semirings. *J. ACM* 71(2):8:1–8:55.

Kirn, J. C.; Meijer, L.; Miltzow, T.; and Bodlaender, H. L. 2025. On Equivalent Characterizations of NP in Abstract Models of Computation. *CoRR* abs/2510.05894.

Kolaitis, P. G.; Pardal, N.; Virtema, J.; and Wijsen, J. 2025. Rewriting consistent answers on annotated data. *Proc. ACM Manag. Data* 3(2):110:1–110:26.

Kostolányi, P. 2024. Weighted automata and logics meet computational complexity. *Inf. Comput.* 301:105213.

Litvinov, G. L.; Rodionov, A. Y.; Sergeev, S. N.; and Sobolevski, A. N. 2013. Universal algorithms for solving the matrix Bellman equations over semirings. *Soft Comput.* 17(10):1767–1785.

Möller, B. 2013. Modal Knowledge and Game Semirings. *Comput. J.* 56(1):53–69.

Muñoz Serrano, T.; Riveros, C.; and Vansummeren, S. 2024. Enumeration and Updates for Conjunctive Linear Algebra Queries Through Expressibility. In Cormode, G., and Shekelyan, M., eds., *27th International Conference on Database Theory (ICDT 2024)*, volume 290 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 12:1–12:20. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Omanovic, A.; Oblak, P.; and Curk, T. 2023. Matrix tri-factorization over the tropical semiring. *IEEE Access* 11:69022–69032.

Peeva, L. 1991. Equivalence, Reduction and Minimization of Finite Automata over Semirings. *Theor. Comput. Sci.* 88(2):269–285.

Poizat, B. 1995. *Les petits cailloux: Une approche modèle-théorique de l'Algorithmie*. Aléas Editeur.

Preparata, F. P., and Shamos, M. I. 1985. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer.

Sipser, M. 1997. *Introduction to the theory of computation*. PWS Publishing Company.

Tannen, V. 2017. Provenance analysis for FOL model checking. *ACM SIGLOG News* 4(1):24–36.

ten Cate, B.; Dalmau, V.; Kolaitis, P. G.; and Wu, W. 2024. When do homomorphism counts help in query algorithms? In Cormode, G., and Shekelyan, M., eds., *27th International Conference on Database Theory, ICDT 2024, Paestum, Italy, March 25-28, 2024*, volume 290 of *LIPIcs*, 8:1–8:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Vrana, P. 2022. A Generalization of Strassen's Theorem on Preordered Semirings. *Order* 39(2):209–228.