Logical Characterizations of Weighted Complexity Classes

Guillermo Badia
The University of Queensland, Brisbane, Australia

Manfred Droste
Leipzig University, Germany

Carles Noguera
University of Siena, Italy

Erik Paul
Leipzig University, Germany

Abstract

Fagin's seminal result characterizing NP in terms of existential second-order logic started the fruitful field of descriptive complexity theory. In recent years, there has been much interest in the investigation of quantitative (weighted) models of computations. In this paper, we start the study of descriptive complexity based on weighted Turing machines over arbitrary semirings. We provide machine-independent characterizations (over ordered structures) of the weighted complexity classes $\mathsf{NP}[\mathcal{S}]$, $\mathsf{FP}[\mathcal{S}]$, $\mathsf{FPLOG}[\mathcal{S}]$, $\mathsf{FPSPACE}[\mathcal{S}]$, and $\mathsf{FPSPACE}_{poly}[\mathcal{S}]$ in terms of definability in suitable weighted logics for an arbitrary semiring $\mathcal{S}$. In particular, we prove weighted versions of Fagin's theorem (even for arbitrary structures, not necessarily ordered, provided that the semiring is idempotent and commutative), the Immerman–Vardi's theorem (originally for P) and the Abiteboul–Vianu–Vardi's theorem (originally for PSPACE). We also discuss a recent open problem proposed by Eiter and Kiesel.

Recently, the above mentioned weighted complexity classes have been investigated in connection to classical counting complexity classes. Furthermore, several classical counting complexity classes have been characterized in terms of particular weighted logics over the semiring $\mathbb{N}$ of natural numbers. In this work, we cover several of these classes and obtain new results for others such as NPMV, $\oplus$P, or the collection of real-valued languages realized by polynomial-time real-valued nondeterministic Turing machines. Furthermore, our results apply to classes based on many other important semirings, such as the max-plus and the min-plus semirings over the natural numbers which correspond to the classical classes $\mathsf{MaxP}[O(\log n)]$ and $\mathsf{MinP}[O(\log n)]$, respectively.

2012 ACM Subject Classification Theory of computation → Logic; Theory of computation → Complexity theory and logic

Keywords and phrases Descriptive complexity, Weighted Turing machines, Weighted logics, Semirings

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.14

Funding *Guillermo Badia*: Australian Research Council grant DE220100544
Carles Noguera: European Union's Marie Sklodowska–Curie grant no. 101007627 (MOSAIC project). GNSAGA, gruppo nazionale di ricerca INdAM

Acknowledgements We are grateful to three anonymous referees for their numerous useful remarks that contributed to improving the presentation of the paper.

1 Introduction

Descriptive complexity is a branch of computational complexity, as well as finite model theory, where the difficulty in solving a problem by a Turing machine is characterized not by the amount of resources required (such as time, space and so on) but rather in terms of the complexity of describing the problem in some logical formalism. This field was initially started in 1974 by Ronald Fagin with the celebrated result in [19] (coined by Neil Immerman as "Fagin's theorem") which stated that the class of NP languages coincides with the class of languages definable in existential second-order logic. Many further surprising results followed this development, particularly the Immerman–Vardi's theorem characterizing P over ordered structures using fixed-point logic [31, 48] and the Abiteboul–Vianu–Vardi characterization of PSPACE in terms of partial fixed-point logic [1, 48]. Today there are several textbooks that cover the fundamentals of the area as a line of research within finite model theory [15, 38, 25, 32]. In this paper, we propose to study quantitative versions of some of these key results in this important field in connection with weighted computation. We work over finite structures that come with a linear ordering, which is a standard restriction in descriptive complexity.

Weighted automata are nondeterministic finite automata augmented with values from a semiring as weights on the transitions [45]. These weights may model, e.g. the cost involved when executing a transition, the amount of resources or time needed for this, or the probability or reliability of its successful execution. The theory of weighted automata and weighted context-free grammars was essential for the solution of such classical automata-theoretic problems as the decidability of the equivalence of unambiguous context-free languages and regular languages [43] (in fact, the only known proofs of this involve weighted automata), the decidability of two given deterministic multitape automata [30], and the decidability of two given deterministic pushdown automata [39, 46]. This led to quick development of this field, described in the books [6, 12, 16, 36, 42, 43]. Furthermore, weighted automata and weighted context-free grammars have been used as basic concepts in natural language processing and speech recognition, as well as in algorithms for digital image compression [2]. Weighted logic [11], with weights in an arbitrary semiring, was developed originally to obtain a weighted version of the Büchi–Elgot–Trakhtenbrot theorem, showing that a certain weighted monadic second-order logic has the same expressive power on words as weighted automata. Consequently, this weighted logic over suitable semirings like fields has similar decidability properties on words as unweighted monadic second-order logic. It is worth remarking that the classical Büchi–Elgot–Trakhtenbrot theorem is usually regarded as part of the "prehistory" of descriptive complexity [25, p. 145].

Weighted Turing machines extend the concept of weighted automata as natural quantitative counterparts of classical Turing machines. They were first introduced under the name "algebraic Turing machines" in [10, 9] and they have attracted further attention in [34]. Instances of this concept include the so called "fuzzy Turing machines" [49, 4]. Recently, the articles [18, 17] have introduced a related notion of "semiring Turing machine" and explicitly asked for the development of descriptive complexity in such framework as an open problem, focusing specifically on Fagin's theorem in connection to weighted logic [18, p. 255]. We will address this problem at the end of Section 5.

Our contribution. The present paper develops a theory of weighted descriptive complexity and establishes quantitative versions of some celebrated classical theorems. The novel contributions of this work can be summarized in the following characterizations (for an arbitrary semiring $\mathcal{S}$):

- The weighted complexity class $\mathsf{NP}[\mathcal{S}]$ coincides with the queries definable by weighted existential second-order logic on ordered structures, with weights in $\mathcal{S}$, respectively for all structures if $\mathcal{S}$ is idempotent and commutative (Theorem 22).
- The weighted complexity class $\mathsf{FP}[\mathcal{S}]$ coincides with the queries definable by weighted inflationary fixed-point logic, with weights in $\mathcal{S}$ (Theorem 26).

- The weighted complexity class $\mathsf{FPSPACE}[\mathcal{S}]$ coincides with the queries definable by weighted partial fixed-point logic with the addition of second-order multiplicative and additive quantifiers, with weights in $\mathcal{S}$ (Theorem 29).
- The weighted complexity class $\mathsf{FPSPACE}_{poly}[\mathcal{S}]$ coincides with the queries definable by weighted partial fixed-point logic, with weights in $\mathcal{S}$ (Theorem 31).
- The weighted complexity class $\mathsf{FPLOG}[\mathcal{S}]$ coincides with the queries definable by weighted deterministic transitive closure logic (Theorem 33).

Related work. Despite the fact that some characterizations of counting complexity classes using Boolean logics were known [33, 44, 14], observe that the article [3] (following up on the work of [44]) already proposes the idea of using certain weighted logics (with weights in the semiring $\mathbb{N}$ of natural numbers or, in a couple of cases, $\mathbb{Z}$) to characterize well-known counting complexity classes. The authors obtain several interesting results that are also covered by our more encompassing work here (that is, they provide logical characterizations of #P, FP, FPSPACE, FPSPACE(*poly*), GapP, and MaxP). There is, however, some orthogonality as they cover some classical complexity classes that we do not and, similarly, we cover some that they do not, as we do not restrict our semiring to being $\mathbb{N}$ or $\mathbb{Z}$. Moreover, the investigation in [3], by contrast to ours, concentrates on the study of classical counting classes for ordered structures, while we consider both ordered and arbitrary structures (provided, in the latter case, that the semiring is idempotent and commutative; examples include e.g. the max-plus- and min-plus-semirings). In the present article, the central aim is rather starting the study of weighted complexity classes via logic, and the corollaries characterizing classical complexity classes are obtained as interesting byproducts of the work. In this way, we are also meeting the challenge posed in [34, p.3] of developing "quantitative descriptive complexity theory based on weighted logics [...] over some fairly general class of semirings". Further work on the model theory of weighted logics includes a Feferman–Vaught result [13], but the area remains largely unexplored despite being one of the open problems suggested in [11]. Finally, an approach related to the weighted logics discussed here has been recently proposed in [26, 27] motivated by problems in database theory [23]. The idea there is that the atomic facts of a model are annotated by values from a semiring whereas in the present paper this aspect is fully classical.

2 Weighted Turing machines

In order to introduce the notion of a weighted Turing machine, first we need to define the kind of algebraic structures that will provide the weights, that is, semirings.

▶ **Definition 1** (Semirings). *A* semiring *is a tuple* $\mathcal{S} = \langle S, +, \cdot, \mathbb{0}, \mathbb{1}\rangle$*, with operations addition* $+$ *and multiplication* $\cdot$ *and constants* $\mathbb{0}$ *and* $\mathbb{1}$ *such that*
- $\langle S, +, \mathbb{0}\rangle$ *is a commutative monoid and* $\langle S, \cdot, \mathbb{1}\rangle$ *is a monoid,*
- *multiplication distributes over addition, and*
- $s \cdot \mathbb{0} = \mathbb{0} \cdot s = \mathbb{0}$ *for every* $s \in S$*.*

We say that $\mathcal{S}$ *is* commutative *if the monoid* $\langle S, \cdot, \mathbb{1}\rangle$ *is commutative, and we say that* $\mathcal{S}$ *is* idempotent *if the monoid* $\langle S, +, \mathbb{0}\rangle$ *is idempotent (that is,* $s + s = s$ *for each* $s \in S$*).*

Some examples of semirings, including those that we will use in this paper, are the following:
- the *Boolean semiring* $\mathbb{B} = \langle \{0, 1\}, \min, \max, 0, 1\rangle$,
- any bounded distributive lattice $\langle L, \vee, \wedge, 0, 1\rangle$,
- the semiring of natural numbers $\langle \mathbb{N}, +, \cdot, 0, 1\rangle$,

- the semiring of extended natural numbers $\langle \mathbb{N} \cup \{+\infty\}, +, \cdot, 0, 1\rangle$ where $0 \cdot (+\infty) = 0$,
- the ring of integers, $\langle \mathbb{Z}, +, \cdot, 0, 1\rangle$,
- the ring of integers modulo n, $\langle \mathbb{Z}_n, +_n, \cdot_n, \overline{0}, \overline{1}\rangle$, for each $n \in \mathbb{N}$,
- the field of rational numbers $\langle \mathbb{Q}, +, \cdot, 0, 1\rangle$,
- the *max-plus* or *arctic semiring* $\mathrm{Arct} = \langle \mathbb{R}_+ \cup \{-\infty\}, \max, +, -\infty, 0\rangle$, where $\mathbb{R}_+$ denotes the set of non-negative real numbers,
- the restriction of the arctic semiring to the natural numbers $\mathbb{N}_{\max} = \langle \mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0\rangle$,
- the *min-plus* or *tropical semiring* $\mathrm{Trop} = \langle \mathbb{R}_+ \cup \{+\infty\}, \min, +, +\infty, 0\rangle$,
- the restriction of the tropical semiring to the natural numbers $\mathbb{N}_{\min} = \langle \mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0\rangle$,
- the semiring $\mathcal{F}_* = \langle [0, 1], \max, *, 0, 1\rangle$ given by a t-norm $*$ [49],
- the semiring of finite languages $2^{\Sigma^*}_{\text{fin}} = \langle 2^{\Sigma^*}_{\text{fin}}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$, for an alphabet Σ,
- the semiring $\mathcal{S}_{\max} = \langle \{0, 1\}^* \cup \{-\infty\}, \max, \cdot, -\infty, \varepsilon\rangle$ of binary words in which max is computed according to the *radix order* (for $x, y \in \{0, 1\}^*$, $x \preceq y$ iff $|x| < |y|$ or $|x| = |y|$ and x is smaller than or equal to y in the lexicographic order) and $\max(x, -\infty) = \max(-\infty, x) = x$ for each x, $\cdot$ is the concatenation operation, and $x \cdot (-\infty) = (-\infty) \cdot x = -\infty$ for each x,
- the semiring $\mathcal{S}_{\min} = \langle \{0, 1\}^* \cup \{+\infty\}, \min, \cdot, +\infty, \varepsilon\rangle$ analogous to the previous one.

▶ **Definition 2** (Weighted Turing Machines)**.** *Let $\mathcal{S}$ be a semiring and Σ an alphabet. A* weighted (or algebraic) Turing machine *over $\mathcal{S}$ and input alphabet Σ is a septuple $\mathcal{M} = \langle Q, \Gamma, \Delta, \nu, q_0, F, \square\rangle$, where*

- *Q is a nonempty finite set whose elements are called* states*,*
- *$\Gamma \supseteq \Sigma$ is an alphabet (*working alphabet*),*
- *$\Delta \subseteq (Q \setminus F) \times \Gamma \times Q \times \Gamma \times \{-1, 0, 1\}$ and its elements are called* transitions*,*
- *$\nu\colon \Delta \longrightarrow S$ is called a* transition weighting function*, $q_0 \in Q$ is called the* initial state*, $F \subseteq Q$ and its elements are called* accepting states*, and $\square \in \Gamma \setminus \Sigma$ is the blank symbol.*

We call $\mathcal{M}$ a Turing machine *if $\mathcal{S}$ is the Boolean semiring $\mathbb{B}$. We call $\mathcal{M}$* deterministic *if for every pair $\langle p, a\rangle \in Q \times \Gamma$, there is at most one transition $\langle p, a, q, b, d\rangle \in \Delta$.*

A *configuration* of $\mathcal{M}$ is a unique description of the machine's state, contents of the working tape, and the position of the machine's head. If $e = \langle p, c, q, d, t\rangle \in \Delta$ is a transition and C_1, C_2 are configurations of $\mathcal{M}$, then we write $C_1 \longrightarrow_e C_2$ if C_1 is a configuration with state p and the head reading c, while C_2 is obtained from C_1 by changing state to q, rewriting the originally read symbol c to d, and moving the head as prescribed by t. We write $C_1 \longrightarrow C_2$ if $C_1 \longrightarrow_e C_2$ for some $e \in \Delta$.

A *computation* of $\mathcal{M}$ is a word $\gamma = C_1 e_1 C_2 e_2 C_3 \dots C_n e_n C_{n+1}$ such that $C_1, \dots, C_{n+1}$ are configurations of $\mathcal{M}$, $e_1, \dots, e_n \in \Delta$, $C_k \longrightarrow_{e_k} C_{k+1}$ for each $k \in \{1, \dots, n\}$, and C_1 is a configuration with state q_0 and the head at the leftmost non-blank cell (if there is some). The *weight* of γ is defined as $\nu(\gamma) := \nu(e_1)\nu(e_2)\dots\nu(e_n)$. γ is called an *accepting* computation if C_{n+1} has an accepting state. We say that γ is a computation on w in Σ^*, and write $\Sigma(\gamma) = w$ if C_1 is a configuration with w on the working tape. We denote the set of all computations of $\mathcal{M}$ by $C(\mathcal{M})$ and the set of all accepting computations by $A(\mathcal{M})$.

▶ **Convention 3.** *From now on we will assume that every Turing machine $\mathcal{M}$ is* finitely terminating*, that is, the set $C_w(\mathcal{M}) = \{\gamma \in C(\mathcal{M}) \mid \Sigma(\gamma) = w\}$ is finite for each $w \in \Sigma^*$. In particular, the set $A_w(\mathcal{M}) = \{\gamma \in A(\mathcal{M}) \mid \Sigma(\gamma) = w\}$ is finite.*

By a *series* σ we mean a mapping $\sigma\colon \Sigma^* \longrightarrow S$ where Σ is an alphabet, Σ^* the corresponding language and $\mathcal{S}$ is a semiring. Thanks to the convention, we can introduce the following notion:

▶ **Definition 4** (Behavior of a weighted Turing machine)**.** *Let $\mathcal{M}$ be a weighted Turing machine. The* behavior *of $\mathcal{M}$ as the mapping $\|\mathcal{M}\|\colon \Sigma^* \longrightarrow S$ defined as*

$$\|\mathcal{M}\|(w) := \sum_{\gamma \in A_w(\mathcal{M})} \nu(\gamma).$$

We say that a series $\sigma\colon \Sigma^* \longrightarrow S$ is recognized *by a weighted Turing machine $\mathcal{M}$ if $\|\mathcal{M}\| = \sigma$.*

The definition of weighted Turing machine we have used here is exactly the same as that of algebraic Turing machines [10, Def. 5.1] (see also [34]). Similarly, the notion of the behavior of the machine coincides. The semiring Turing machines of [17, 18], by contrast, differ in that they impose some conditions on the allowed transitions [18, cf. Def. 12]. Given distributivity of multiplication over addition, the notion of a semiring Turing machine function in [18, Def. 13] coincides with that of the behavior we use here. Semiring Turing Machines allow semiring values on the tape in somewhat of a black-box manner. Intuitively, one can transition with the weight of the value on the tape, but cannot differentiate the values on the tape or modify them. If semiring values are not allowed in the input string then the definition of weighted and semiring Turing Machines are equivalent (in the sense that one can be transformed into the other without a significant change of execution time). All these definitions generalize the corresponding notions for weighted automata.

3 Some weighted complexity classes

Let $\mathcal{M} = \langle Q, \Gamma, \Delta, \nu, q_0, F, \Box\rangle$ be a weighted Turing machine over $\mathcal{S}$ and Σ. For $w \in \Sigma^*$, we denote by $\mathsf{TIME}(\mathcal{M}, w)$ the maximal length of a computation of $\mathcal{M}$ on w, and define, for $n \in \mathbb{N}$, $\mathsf{TIME}(\mathcal{M}, n) := \max\{\mathsf{TIME}(\mathcal{M}, w) : w \in \Sigma^*, |w| \leq n\}$.

For a function $f\colon \mathbb{N} \longrightarrow \mathbb{N}$, we denote by $\mathsf{SERIES}[S, \Sigma](f)$ the set of all series σ such that $\sigma = \|\mathcal{M}\|$ for some weighted Turing machine $\mathcal{M}$ over $\mathcal{S}$ and Σ with $\mathsf{TIME}(\mathcal{M}, n) = O(f(n))$. Now we can define the complexity classes:

$\mathsf{SERIES}[\mathcal{S}](f(n)) := \bigcup\{\mathsf{SERIES}[S, \Sigma](f(n)) : \Sigma \text{ is an alphabet}\}$.

▶ **Definition 5.** *Let $\mathcal{S}$ be a semiring. We define the following weighted complexity class*

$$\mathsf{NP}[\mathcal{S}] := \bigcup\{\mathsf{SERIES}[\mathcal{S}](n^k) : k \in \mathbb{N}\}.$$

$\mathsf{NP}[\mathcal{S}]$ (cf. [34, Def. 4.1]) coincides with the definition of the class $\mathcal{S}$-#P in [10, Def. 5.2]. Furthermore, it is contained as a subclass in the similarly defined class NP[$\mathcal{R}$] from [18, Def. 14] when $\mathcal{R}$ is a commutative semiring. Below (Proposition 35), we will actually show that this containment is proper, in the sense that NP[$\mathcal{R}$] will contain some series that are not in $\mathsf{NP}[\mathcal{S}]$.

▶ **Example 6.** Following [10, Prop. 5.3] and [34, Examples 4.2–4.6], we can list some prominent instances of $\mathsf{NP}[\mathcal{S}]$:

- the usual complexity class NP, obtained when $\mathcal{S} = \mathbb{B}$ is the two-element Boolean semiring and each transition is weighted by 1 (this is the standard way of representing a classical machine model in the weighted context),
- the counting class #P [47], obtained when $\mathcal{S} = \langle \mathbb{N}, +, \cdot, 0, 1\rangle$ is the semiring of natural numbers and each transition is weighted by 1,

- the complexity class $\bigoplus\mathsf{P}$ [40], obtained when $\mathcal{S} = \langle \mathbb{Z}_2, +_2, \cdot_2, \overline{0}, \overline{1}\rangle$ is the finite field of two elements and each transition is weighted by 1,
- the class GapP, closure of $\#\mathsf{P}$ under subtraction [20, 28], obtained when $\mathcal{S} = \langle \mathbb{Z}, +, \cdot, \overline{0}, \overline{1}\rangle$ is the ring of integers and transitions are weighted by 1 and -1,
- the class $\mathsf{MOD}_q - \mathsf{P}$ (for $q \geq 2$) [8], defined similarly to $\#\mathsf{P}$ but with respect to counting modulo q, obtained when $\mathcal{S} = \langle \mathbb{Z}_q, +_q, \cdot_q, \overline{0}, \overline{1}\rangle$ and transitions are weighted by 1.

▶ **Example 7.** Some further instances of $\mathsf{NP}[\mathcal{S}]$, this time following [34, Examples 4.7–4.11], are:

- the class $\mathsf{NP}[F_*]$ of all fuzzy languages realizable by fuzzy Turing machines [49] with t-norm $*$ in polynomial time, obtained when the semiring is $\mathcal{F}_* = \langle [0, 1], \max, *, 0, 1\rangle$ and the weights correspond to degrees of membership in the fuzzy language,
- the class NPMV of all multivalued functions realized by nondeterministic polynomial-time transducer machines [7], obtained when, given alphabets Σ_1 and Σ_2, the semiring is $\langle 2^{\Sigma_2^*}_{\text{fin}}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ and weighted Turing machines have input alphabet Σ_1,
- the class of all multiset-valued functions computed by nondeterministic polynomial-time transducer machines with counting, obtained as in the previous example but using the free semiring $\langle \mathbb{N}\langle\Sigma_2^*\rangle, +, \cdot, 0, 1\rangle$ instead,
- the class $\mathsf{MaxP} \subseteq \mathsf{OptP}$ of problems in which the objective is to compute the value of a solution to an optimization problem in NPO [35], obtained when the semiring is $\mathcal{S}_{\max}$, and the class $\mathsf{MinP} \subseteq \mathsf{OptP}$, obtained when the semiring is $\mathcal{S}_{\min}$,
- the class $\mathsf{MaxP}[[O(\log n)]] \subseteq \mathsf{OptP}[O(\log n)]$ of problems in which the objective is to compute the value of a solution to an optimization problem in $\mathsf{NPO\,PB}$ [35], obtained when the semiring is $\mathbb{N}_{\max}$, and $\mathsf{MinP}[[O(\log n)]] \subseteq \mathsf{OptP}[O(\log n)]$, , obtained when the semiring is $\mathbb{N}_{\min}$.

A notion from universal algebra (cf. [5]) that we will make use of in defining some of the complexity classes below (e.g. $\mathsf{FP}[\mathcal{S}], \mathsf{FPSPACE}[\mathcal{S}]$ and $\mathsf{FPLOG}[\mathcal{S}]$) is the following:

▶ **Definition 8** (Term algebra)**.** *Consider a semiring $\mathcal{S} = \langle S, +, \cdot, \mathbb{0}, \mathbb{1}\rangle$ and a subset $X \subseteq S$. The* set of terms *$T(X)$ is the collection of all well-formed strings that can be constructed using the symbols in X and $+', \cdot', \mathbb{0}, \mathbb{1}$ (in particular, $\mathbb{0}, \mathbb{1} \in T(X)$), that is, the smallest set such that: (1) $X \subseteq T(X)$ and (2) $(t_1 +' t_2) \in T(X)$ and $(t_1 \cdot' t_2) \in T(X)$ for every two terms $t_1, t_2 \in T(X)$; we abuse notation and omit parentheses whenever associativity permits. The* term algebra *$\mathcal{T}(X)$ is the structure with universe $T(X)$ and operations $+', \cdot'$ defined in the obvious way.*

Recall that classically FP is the set of function problems that can be solved by a deterministic Turing machine in polynomial time.

▶ **Definition 9.** *We define the complexity class $\mathsf{FP}[\mathcal{S}]$ as*

$$\mathsf{FP}[\mathcal{S}] := \bigcup_{\substack{\{0,1\}\subseteq G \subseteq_{\text{fin}} S \\ \Sigma \text{ is a finite alphabet}}} \mathsf{FP}[G, \Sigma]$$

where $\mathsf{FP}[G, \Sigma]$ is the set of all series $\sigma\colon \Sigma^* \longrightarrow \langle G\rangle$ (where $\langle G\rangle$ is the subsemiring of $\mathcal{S}$ generated by G) such that there is a constant $k \in \mathbb{N}$ and a deterministic polynomial-time Turing machine which outputs for every word $w \in \Sigma^*$ a word of the form $\sum_{i_1=1}^{m_1} \prod_{j_1=1}^{n_1} \cdots \sum_{i_k=1}^{m_k} \prod_{j_k=1}^{n_k} s_{i_1 j_1 \cdots i_k j_k}$ in the algebra of terms $T(G)$ in S with value $\sigma(w)$ in $\mathcal{S}$.

In Definition 9, we employ a classical deterministic Turing machine which outputs, in each transition, symbols from $G \cup \{(,), +', \cdot', \mathbb{0}, \mathbb{1}\}$ or a blank. Thus, for our outputs we could obtain arbitrarily complex expressions. Therefore, the constant k limiting the number of alternations of sums and products is a proper restriction. Hence this definition of $\mathsf{FP}[\mathcal{S}]$ differs from the one of [34]. Later on, we will model logical formulas with alternating sum and product quantifiers using Turing machines which compute functions in $\mathsf{FP}[\mathcal{S}]$, hence $k = 1$ would be insufficient to model these alternations. For the converse, in order to model these Turing machines by formulas, the number of alternations of sums and products in each such Turing machine needs to be bounded to obtain a formula with nested quantifiers.

▶ **Example 10.** If $\mathcal{S} = \mathbb{B}$ is the two-element Boolean semiring, then $\mathsf{FP}[\mathbb{B}]$ is just P [34, Example 5.4]. Observe that the terms output by the machine in that example are already trivially of the form $\sum_{i=1}^{n} \prod_{j=1}^{m} s_{ij}$.

FP is to $\#\mathsf{P}$ what P is to NP. Thus, considering $\mathsf{NP}[\mathcal{S}]$ as a generalization of $\#\mathsf{P}$ (as it is done in [10]), the relationship between $\mathsf{FP}[\mathcal{S}]$ and $\mathsf{NP}[\mathcal{S}]$ is similar to that between P and NP.

▶ **Example 11.** If $\mathcal{S} = \mathbb{N}$ is the natural numbers semiring, then $\mathsf{FP}[\mathbb{N}]$ is just FP [34, Example 5.5]. As before, observe that the terms output by the machine in that example are already of the form $\sum_{i=1}^{n} \prod_{j=1}^{m} s_{ij}$.

▶ **Definition 12.** *The class* $\mathsf{FPLOG}[\mathcal{S}]$ *is defined as* $\mathsf{FP}[\mathcal{S}]$ *except that we allow the machine to have logarithmic space on the length of the input rather than polynomial time.*

▶ **Example 13.** If $\mathcal{S} = \mathbb{B}$, then $\mathsf{FPLOG}[\mathbb{B}]$ is just $\mathsf{DLOGSPACE}$.

▶ **Example 14.** If $\mathcal{S} = \mathbb{N}$, then $\mathsf{FPLOG}[\mathbb{N}]$ is just FPLOG, which is defined as FP but allowing the machine to use logarithmic space on the size of the input (cf. [22]).

▶ **Definition 15.** *The class* $\mathsf{FPSPACE}[\mathcal{S}]$ *is defined as* $\mathsf{FP}[\mathcal{S}]$ *except that we allow the machine to have polynomial space on the length of the input rather than polynomial time.*

▶ **Example 16.** If $\mathcal{S} = \mathbb{B}$, then $\mathsf{FPSPACE}[\mathbb{B}]$ is just PSPACE.

▶ **Example 17.** If $\mathcal{S} = \mathbb{N}$, then $\mathsf{FPSPACE}[\mathbb{N}]$ is just $\mathsf{FPSPACE}$ ([37]).

▶ **Definition 18.** *The class* $\mathsf{FPSPACE}_{poly}[\mathcal{S}]$ *is defined as* $\mathsf{FPSPACE}[\mathcal{S}]$ *except that we require the word* $\sum_{i=1}^{n} \prod_{j=1}^{m} s_{ij}$ *to have length bounded by a polynomial. Here, every semiring element is considered to have length* 1*.*

▶ **Example 19.** If $\mathcal{S} = \mathbb{B}$, then $\mathsf{FPSPACE}_{poly}[\mathbb{B}]$ is just PSPACE.

▶ **Example 20.** If $\mathcal{S} = \mathbb{N}$, then $\mathsf{FPSPACE}_{poly}[\mathbb{N}]$ is just $\mathsf{FPSPACE}_{poly}$ ([37]).

4 Weighted logics

A *vocabulary* (or *signature*) τ is a pair $\langle \mathrm{Rel}_\tau, \mathrm{ar}_\tau \rangle$ where Rel_τ is a set of relation symbols and $\mathrm{ar}_\tau \colon \mathrm{Rel}_\tau \longrightarrow \mathbb{N}_+$ is the arity function. A *τ-structure* $\mathfrak{A}$ is a pair $\langle A, \mathcal{I}_\mathfrak{A} \rangle$ where A is a set, called the *universe of* $\mathfrak{A}$, and $\mathcal{I}_\mathfrak{A}$ is an *interpretation*, which maps every symbol $R \in \mathrm{Rel}_\tau$ to a set $R^\mathfrak{A} \subseteq A^{\mathrm{ar}_\tau(R)}$. We assume that each structure is *finite*, that is, its universe is a finite set. A structure is called *ordered* if it is given for a vocabulary $\tau \cup \{<\}$ where $<$ is interpreted as a linear ordering with endpoints. By $\mathrm{Str}(\tau)_<$ we denote the class of all finite ordered τ-structures.

We provide a countable set $\mathcal{V}$ of first and second-order variables, where lower case letters like x and y denote first-order variables and capital letters like X and Y denote second-order variables. Each second-order variable X comes with an associated arity, denoted by $\mathrm{ar}(X)$. We define first-order formulas β over a signature τ and weighted first-order formulas φ over τ and a semiring $\mathcal{S}$, respectively, by the grammars

$$\beta ::= \texttt{false} \mid R(x_1,\dots,x_n) \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\beta$$
$$\varphi ::= \beta \mid s \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi,$$

where $R \in \mathrm{Rel}_\tau$, $n = \mathrm{ar}_\tau(R)$, $x, x_1, \dots, x_n \in \mathcal{V}$ are first-order variables, and $s \in S$. Likewise, we define second-order formulas β over τ and weighted second-order formulas φ over τ and $\mathcal{S}$ through

$$\beta ::= \texttt{false} \mid R(x_1,\dots,x_n) \mid X(x_1,\dots,x_n) \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\beta \mid \exists X.\beta$$
$$\varphi ::= \beta \mid s \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes X.\varphi,$$

with $R \in \mathrm{Rel}_\tau$, $n = \mathrm{ar}_\tau(R) = \mathrm{ar}(X)$, $x, x_1, \dots, x_n \in \mathcal{V}$ first-order variables, $X \in \mathcal{V}$ a second-order variable, and $s \in S$. We also allow the usual abbreviations $\wedge$, $\forall$, $\rightarrow$, $\leftrightarrow$, and `true`. By $\mathrm{FO}(\tau)$ and $\mathrm{wFO}(\tau, S)$ we denote the sets of all first-order formulas over τ and all weighted first-order formulas over τ and $\mathcal{S}$, respectively, and by $\mathrm{SO}(\tau)$ and $\mathrm{wSO}(\tau, S)$ we denote the sets of all second-order formulas over τ and all weighted second-order formulas over τ and $\mathcal{S}$, respectively.

The notion of *free variables* is defined as usual, i.e., the operators $\exists, \forall, \bigoplus$, and $\bigotimes$ bind variables. We let $\mathrm{Free}(\varphi)$ be the set of all free variables of φ. A formula φ with $\mathrm{Free}(\varphi) = \emptyset$ is called a *sentence*. For a tuple $\bar{\varphi} = \langle \varphi_1, \dots, \varphi_n \rangle \in \mathrm{wSO}(\tau, S)^n$, we define $\mathrm{Free}(\bar{\varphi}) = \bigcup_{i=1}^n \mathrm{Free}(\varphi_i)$.

We define the semantics of SO and wSO as follows. Let τ be a signature, $\mathfrak{A} = \langle A, \mathcal{I}_\mathfrak{A} \rangle$ a τ-structure, and $\mathcal{V}$ a set of first and second-order variables. A $(\mathcal{V}, \mathfrak{A})$-assignment ρ is a function $\rho\colon \mathcal{V} \longrightarrow A \cup \mathcal{P}(A)$ such that, whenever $x \in \mathcal{V}$ is a first-order variable and $\rho(x)$ is defined, we have $\rho(x) \in A$, and whenever $X \in \mathcal{V}$ is a second-order variable and $\rho(X)$ is defined, we have $\rho(X) \subseteq A^{\mathrm{ar}(X)}$. For a first-order variable, this restriction may cause the variable to become undefined. Let $\mathrm{dom}(\rho)$ be the domain of ρ. For a first-order variable $x \in \mathcal{V}$ and an element $a \in A$, the *update* $\rho[x \to a]$ is defined through $\mathrm{dom}(\rho[x \to a]) = \mathrm{dom}(\rho) \cup \{x\}$, $\rho[x \to a](\mathcal{X}) = \rho(\mathcal{X})$ for all $\mathcal{X} \in \mathcal{V} \setminus \{x\}$, and $\rho[x \to a](x) = a$. For a second-order variable $X \in \mathcal{V}$ and a set $I \subseteq A^{\mathrm{ar}(X)}$, the update $\rho[X \to I]$ is defined in a similar fashion. By $\mathfrak{A}_\mathcal{V}$ we denote the set of all $(\mathcal{V}, \mathfrak{A})$-assignments.

For $\rho \in \mathfrak{A}_\mathcal{V}$ and a formula $\beta \in \mathrm{SO}(\tau)$ the relation "$\langle \mathfrak{A}, \rho \rangle$ satisfies β", denoted by $\langle \mathfrak{A}, \rho \rangle \models \beta$, is defined as

$$
\begin{aligned}
&\langle \mathfrak{A}, \rho \rangle \models \texttt{false} && \text{never holds}\\
&\langle \mathfrak{A}, \rho \rangle \models R(x_1,\dots,x_n) &\iff& x_1,\dots,x_n \in \mathrm{dom}(\rho) \text{ and } (\rho(x_1),\dots,\rho(x_n)) \in R^\mathfrak{A}\\
&\langle \mathfrak{A}, \rho \rangle \models X(x_1,...,x_n) &\iff& x_1,...,x_n, X \in \mathrm{dom}(\rho) \text{ and } \langle \rho(x_1),\dots,\rho(x_n) \rangle \in \rho(X)\\
&\langle \mathfrak{A}, \rho \rangle \models \neg\beta &\iff& \langle \mathfrak{A}, \rho \rangle \models \beta \text{ does not hold}\\
&\langle \mathfrak{A}, \rho \rangle \models \beta_1 \vee \beta_2 &\iff& \langle \mathfrak{A}, \rho \rangle \models \beta_1 \text{ or } \langle \mathfrak{A}, \rho \rangle \models \beta_2\\
&\langle \mathfrak{A}, \rho \rangle \models \exists x.\beta &\iff& \langle \mathfrak{A}, \rho[x \to a] \rangle \models \beta \text{ for some } a \in A\\
&\langle \mathfrak{A}, \rho \rangle \models \exists X.\beta &\iff& \langle \mathfrak{A}, \rho[X \to I] \rangle \models \beta \text{ for some } I \subseteq A.
\end{aligned}
$$

Let $\varphi \in \mathrm{wSO}(\tau, S)$ and $\mathfrak{A} \in \mathrm{Str}(\tau)_<$, $a_1, \dots, a_k$ be an enumeration of the elements of $\mathfrak{A}$ according to the ordering that serves as the interpretation of $<$, and for every integer n, let $I_1^n, \dots, I_{l_n}^n$ be an enumeration of the subsets of A^n according to the lexicographic ordering induced by the interpretation of $<$. The *(weighted) semantics* of φ is a mapping $[\![\varphi]\!](\mathfrak{A}, \cdot) \colon \mathfrak{A}_\mathcal{V} \longrightarrow S$ inductively defined as

$$
\begin{aligned}
[\![\beta]\!](\mathfrak{A}, \rho) &= \begin{cases} \mathbb{1} & \text{if } \langle \mathfrak{A}, \rho \rangle \models \beta \\ \mathbb{0} & \text{otherwise} \end{cases} \\
[\![s]\!](\mathfrak{A}, \rho) &= s \\
[\![\varphi_1 \oplus \varphi_2]\!](\mathfrak{A}, \rho) &= [\![\varphi_1]\!](\mathfrak{A}, \rho) + [\![\varphi_2]\!](\mathfrak{A}, \rho) \\
[\![\varphi_1 \otimes \varphi_2]\!](\mathfrak{A}, \rho) &= [\![\varphi_1]\!](\mathfrak{A}, \rho) \cdot [\![\varphi_2]\!](\mathfrak{A}, \rho) \\
[\![\bigoplus x.\varphi]\!](\mathfrak{A}, \rho) &= \sum_{a \in A} [\![\varphi]\!](\mathfrak{A}, \rho[x \to a]) \\
[\![\bigotimes x.\varphi]\!](\mathfrak{A}, \rho) &= \prod_{1 \le i \le k} [\![\varphi]\!](\mathfrak{A}, \rho[x \to a_i]) \\
[\![\bigoplus X.\varphi]\!](\mathfrak{A}, \rho) &= \sum_{I \subseteq A^{\mathrm{ar}(X)}} [\![\varphi]\!](\mathfrak{A}, \rho[X \to I]) \\
[\![\bigotimes X.\varphi]\!](\mathfrak{A}, \rho) &= \prod_{1 \le i \le l_{\mathrm{ar}(X)}} [\![\varphi]\!](\mathfrak{A}, \rho[X \to I_i^{\mathrm{ar}(X)}]).
\end{aligned}
$$

Thanks to the lexicografic ordering our product quantifiers have a well-defined semantics. Note that if the semiring is commutative, in the clauses of universal quantifiers, the semantics is defined by using any order for the factors in the products.

We will usually identify a pair $\langle \mathfrak{A}, \emptyset \rangle$ (where $\emptyset$ is the empty mapping) with $\mathfrak{A}$. We will also refer to the following expansions of FO:

- Transitive closure logic (TC) is obtained by adding the following rule for building formulas: if $\varphi(\overline{x}, \overline{y})$ is a formula with variables $\overline{x} = x_1, \dots, x_k$ and $\overline{y} = y_1, \dots, y_k$, and $\overline{u}, \overline{v}$ are k-tuples of terms, then $[\mathbf{tc}_{\overline{x},\overline{y}}\, \varphi(\overline{x}, \overline{y})](\overline{u}, \overline{v})$ is also a formula, and its semantics is given as $\mathfrak{A} \models [\mathbf{tc}_{\overline{x},\overline{y}}\, \varphi(\overline{x}, \overline{y})](\overline{a}, \overline{b}) \iff$ there exist an $n \ge 1$ and $\overline{c_0}, \dots, \overline{c_n} \in A^k$ such that $\overline{c_0} = \overline{a}$, $\overline{c_n} = \overline{b}$, and $\mathfrak{A} \models \varphi(\overline{c_i}, \overline{c_{i+1}})$ for each $i \in \{0, \dots, n-1\}$.
- Deterministic transitive closure logic (DTC) is obtained by adding the following rule for building formulas: if $\varphi(\overline{x}, \overline{y})$ is a formula with variables $\overline{x} = x_1, \dots, x_k$ and $\overline{y} = y_1, \dots, y_k$, and $\overline{u}, \overline{v}$ are k-tuples of terms, then $[\mathbf{dtc}_{\overline{x},\overline{y}}\, \varphi(\overline{x}, \overline{y})](\overline{u}, \overline{v})$ is also a formula, and its semantics is defined by the equivalence $[\mathbf{dtc}_{\overline{x},\overline{y}}\, \varphi(\overline{x}, \overline{y})](\overline{u}, \overline{v}) \equiv [\mathbf{tc}_{\overline{x},\overline{y}}\, \varphi(\overline{x}, \overline{y}) \wedge \forall z(\varphi(\overline{x}, \overline{z}) \to \overline{y} = \overline{z})](\overline{u}, \overline{v})$.
- Least fixed-point logic (LFP) is obtained by adding the following rules for building formulas: if $\varphi(R, \overline{x})$ is a formula of vocabulary $\tau \cup \{R\}$ with only positive occurrences of R, $\overline{x}$ is a tuple of variables, and $\overline{t}$ is a tuple of terms (both matching the arity of R), then $[\mathbf{lfp}\, R\overline{x}.\psi](\overline{t})$ and $[\mathbf{gfp}\, R\overline{x}.\psi](\overline{t})$ are also formulas. For their semantics, we need to define some auxiliary notions. The *update operator* $F_\psi \colon \mathcal{P}(A^k) \longrightarrow \mathcal{P}(A^k)$ is defined by $F_\psi(R) := \{\overline{a} \mid \langle \mathfrak{A}, R \rangle \models \psi(R, \overline{a})\}$ for any relation R, and it is monotone because R occurs only positively in ψ. A *fixed point* of F_ψ is a relation R such that $F_\psi(R) = R$. Since F_ψ is monotone, it has a least and a greatest fixed point (by Knaster–Tarski Theorem). The semantics is given by: $\mathfrak{A} \models [\mathbf{lfp}\, R\overline{x}.\psi](\overline{t})$ iff $\overline{t}^{\mathfrak{A}}$ is contained in the least fixed point of F_ψ (analogously for $[\mathbf{gfp}\, R\overline{x}.\psi](\overline{t})$ and the greatest fixed point).
- Partial fixed-point logic (PFP) is obtained by adding the following rule for building formulas: if $\varphi(R, \overline{x})$ is a formula of vocabulary $\tau \cup \{R\}$, $\overline{x}$ is a tuple of variables, and $\overline{t}$ is a tuple of terms (both matching the arity of R), then $[\mathbf{pfp}\, R\overline{x}.\psi](\overline{t})$ is also a formula. For

the semantics, we consider again the update operator (now not necessarily monotone) and the sequence of its finite stages: $R^0 := \emptyset$ and $R^{m+1} := F_\psi(R^m)$. In a finite structure $\mathfrak{A}$, the sequence either reaches a fixed point or it enters a cycle of period greater than one. We define the partial fixed point of F_ψ as the fixed point reached in the former case, or as the empty set in the latter case. Now, the semantics is given by: $\mathfrak{A} \models [\mathbf{pfp}\, R\overline{x}.\psi](\overline{t})$ iff $\overline{t}^{\mathfrak{A}}$ is contained in the partial fixed point of F_ψ.

- Inflationary fixed-point logic (IFP) is obtained by adding the following rules for building formulas: if $\varphi(R, \overline{x})$ is a formula of vocabulary $\tau \cup \{R\}$, $\overline{x}$ is a tuple of variables, and $\overline{t}$ is a tuple of terms (both matching the arity of R), then $[\mathbf{ifp}\, R\overline{x}.\psi](\overline{t})$ is also a formula. For its semantics, we need to define some auxiliary notions. An operator $G : \mathcal{P}(B) \longrightarrow \mathcal{P}(B)$ is said to be *inflationary* if $X \subseteq G(X)$ for all $X \in \mathcal{P}(B)$. With any operator $F : \mathcal{P}(B) \longrightarrow \mathcal{P}(B)$ one can associate an inflationary operator G by setting $G(X) := X \cup F(X)$. Iterating G gives a fixed point that we will called the *inflationary fixed point* of F. The semantics is given by: $\mathfrak{A} \models [\mathbf{ifp}\, R\overline{x}.\psi](\overline{t})$ iff $\overline{t}^{\mathfrak{A}}$ is contained in the inflationary fixed point of F_ψ.

The weighted version of each of these logics is defined analogously as in the case of FO and SO by expanding the logics TC, DTC, LFP, PFP, and IFP with the same weighted constructs as given for wFO and wSO. By a famous result of Gurevich and Shelah [29], on finite structures, LFP coincides with IFP and thus their weighted versions, wIFP and wLFP, as we have defined them here, will also coincide in expressive power.

5 Logical characterizations of complexity classes

We are finally ready to present and prove the main results of the paper: the quantitative versions of several logical characterizations of prominent complexity classes. We may assume that every $\mathfrak{A} \in \mathrm{Str}(\tau)_<$ is encoded by a string of 0s and 1s. For example, where $\mathfrak{A} = \langle A, R_1^{\mathfrak{A}}, \ldots, R_j^{\mathfrak{A}} \rangle$ with $|A| = n$ (and we may assume in fact that $A = \{0, \ldots, n-1\}$) we might let

$$\mathrm{enc}(\mathfrak{A}) = \mathrm{enc}(R_1^{\mathfrak{A}}) \cdot \cdots \cdot \mathrm{enc}(R_j^{\mathfrak{A}})$$

where if $R_i^{\mathfrak{A}}$ is an l-ary relation, then $\mathrm{enc}(R_i^{\mathfrak{A}})$ is a string of symbols of length n^l with a 1 in its mth position if the mth tuple of n^l is in $R_i^{\mathfrak{A}}$ and a 0 otherwise.

▶ **Definition 21.** *Consider a weighted logic* $\mathrm{L}[\mathcal{S}]$ *(with weights in a semiring* $\mathcal{S}$*) and a weighted complexity class* $\mathcal{C}$*, which is simply a collection of series. We say that* $\mathrm{L}[\mathcal{S}]$ captures $\mathcal{C}$ *over ordered structures in the vocabulary* $\tau = \{R_1, \ldots, R_j\}$ *if:*

(1) *For every* $\mathrm{L}[\mathcal{S}]$*-formula* ϕ*, there exists* $P \in \mathcal{C}$ *such that* $P(enc(\mathfrak{A})) = \|\phi\|(\mathfrak{A})$ *for every finite ordered* τ*-structure* $\mathfrak{A}$*, and*

(2) *For every* $P \in \mathcal{C}$*, there exists an* $\mathrm{L}[\mathcal{S}]$*-formula* ϕ *such that* $P(enc(\mathfrak{A})) = \|\phi\|(\mathfrak{A})$ *for every finite ordered* τ*-structure* $\mathfrak{A}$*.*

The seminal Fagin's Theorem characterizes NP for ordered structures by existential second-order logic. Our goal is to present a weighted version of this result with arbitrary semirings as weight structures. Whereas in the classical setting one obtains an equivalence between the existence of runs of a Turing machine vs. the satisfiability of an existential logical formula, in the weighted setting of general semirings we have to derive a one-to-one correspondence between the runs of a Turing machine and satisfying assignments for the formulas. Moreover, due to the absence of a natural negation function in the semiring, here, beyond the classical setting, we need conjunctions and universal quantifications. For weighted finite automata over words, in [11] weighted conjunction and universal quantification turned out to be too powerful in general and had to be restricted. Surprisingly, here we do not need these restrictions, but we can show the expressive equivalence between weighted polynomial-time Turing machines and the full weighted existential second-order logic. Moreover, we do not need commutativity of the multiplication of $\mathcal{S}$ (essential in [11]), but can develop our characterization for arbitrary, also non-commutative, semirings $\mathcal{S}$. This is due to new constructions, in this setting, for the involved weighted Turing machines. By wESO we mean the fragment of wSO where the only second-order quantifiers appear at the beginning of the formula and are additive existential.

▶ **Theorem 22** (Weighted Fagin's theorem). *Let $\mathcal{S}$ be a semiring and τ a vocabulary.*
(i) *The logic* wESO$[\mathcal{S}]$ *captures* NP$[\mathcal{S}]$ *over* ordered *finite τ-structures.*
(ii) *Assume that $\mathcal{S}$ is idempotent and commutative. Then, the logic* wESO$[\mathcal{S}]$ *captures* NP$[\mathcal{S}]$ *over* all *finite structures in the vocabulary τ.*

Let us indicate some ideas for the proof. For (i), first, for a given wESO-formula ϕ, we have to construct an NP Turing machine $\mathcal{M}$ with $\|\phi\| = \|\mathcal{M}\|$. For Boolean formulas β, we can follow the classical proof. Regarding weighted formulas ϕ, let us comment on the interesting cases. For weighted conjunctions and universal quantifications, we employ new constructions. Since we are dealing with Turing machines, we can execute weighted Turing machines for the components successively, by saving the word and using transitions of weight 1 in a deterministic way to restore the initial tape configuration. We can show, using the distributivity of the semiring, that the constructed nondeterministic machine $\mathcal{M}$ computes precisely the values prescribed by the semantics of the weighted conjunction or the weighted universal quantifications, respectively.

Second, given a weighted NP Turing machine $\mathcal{M}$, by the assumption on its polynomial time usage, we construct a second-order formula ψ reflecting the accepting computation paths of $\mathcal{M}$ and their employed transitions in a one-to-one correspondence; this enables us to incorporate the weights of the transitions by means of constants in the formula. The order is used for the construction of the formula such that the interpretation of weighted universal quantification reflects precisely the weights of the computation sequences of the given Turing machine.

For (ii), the order in universal quantifications now is taken care of by the commutativity of the multiplication, and the existence of an order is taken care of by an additional existential second-order quantification where idempotency of $\mathcal{S}$ implies that we obtain the same value.

From Theorem 22 and Examples 6 and 7, we immediately obtain the following corollary:

▶ **Corollary 23.** *For ordered structures in a finite vocabulary τ, we have that:*
(1) wESO$[\mathbb{B}]$ captures NP *(originally proved in [19]).*
(2) wESO$[\mathbb{N}]$ captures #P *(originally proved in [3] and [44]).*
(3) wESO$[\mathbb{Z}]$ captures GapP *(originally proved in [3]).*
(4) wESO$[\mathcal{S}_{\max}]$ *(respectively,* wESO$[\mathcal{S}_{\min}]$*).* captures MaxP *(*MinP*) (originally proved in [3]).*
(5) wESO$[\mathbb{Z}_2]$ captures $\bigoplus$P.
(6) wESO$[\mathbb{Z}_q]$ captures MOD$_q$ − P.
(7) wESO$[\mathbb{N}_{\max}]$ *(respectively,* wESO$[\mathbb{N}_{\min}]$*)* captures MaxP$[O(\log n)]$ *(*MinP$[O(\log n)]$*).*
(8) wESO$[\mathcal{F}_*]$ captures *the class of all fuzzy languages realizable by fuzzy Turing machines with t-norm $*$ in polynomial time.*
(9) wESO$[2^{\Sigma_2^*}_{\mathrm{fin}}]$ captures NPMV.
(10) wESO$[\mathbb{N}\langle\Sigma_2^*\rangle]$ captures *the class of all multiset-valued functions computed by non-deterministic polynomial-time transducer machines with counting.*

▶ Remark 24. It is worth observing that the proofs of (2)-(4) in [3] (Prop. 4.2, Cor. 4.8, and Thm. 4.10) are (as expected) different from ours. Our argument works in all those cases but neither of the three arguments given in [3] works for our more general setting.

Our next application of the weighted Fagin's theorem consist in providing a natural computational problem complete for the class $\mathsf{NP}[\mathcal{S}]$ for certain semirings $\mathcal{S}$. Given a semiring $\mathcal{S}$, alphabets Σ_1, Σ_2, and series $\sigma_1 : \Sigma_1^* \longrightarrow S$ and $\sigma_2 : \Sigma_2^* \longrightarrow S$, we say that σ_1 is *polynomially many-one reducible* to σ_2 ($\sigma_1 \leq_m \sigma_2$, in symbols) if there is an $f : \Sigma_1^* \longrightarrow \Sigma_2^*$ computable deterministically in polynomial time such that $\sigma_2(f(w)) = \sigma_1(w)$ for each $w \in \Sigma_1^*$. A series $\sigma : \Sigma^* \longrightarrow S$ is said to be $\mathsf{NP}[\mathcal{S}]$-*hard* if $\sigma' \leq_m \sigma$ for all σ' in $\mathsf{NP}[\mathcal{S}]$. If, moreover, σ belongs to $\mathsf{NP}[\mathcal{S}]$, then it is called $\mathsf{NP}[\mathcal{S}]$-*complete*.

Fix an infinite set X. The language of the weighted propositional logic over a finitely generated semiring $\mathcal{S}$ is built from X as propositional variables, elements of S as truth-constants, and logical connectives $\wedge, \vee, \neg$ (where negation is only applied to propositional variables). Let $\mathtt{Fmla}[\mathcal{S}]$ be the set of all formulas.

A *truth assignment* is a mapping $V : X \longrightarrow \{0, 1\}$ extended to $\overline{V}$ for all formulas in the following way:

1. For each propositional variable $x \in X$, let $\overline{V}(x) := V(x)$ and $\overline{V}(\neg x) := 1$ iff $V(x) = 0$. Moreover, let $\overline{V}(a) := a$ for each $a \in S$.
2. $\overline{V}(\varphi \vee \psi) := \overline{V}(\varphi) + \overline{V}(\psi)$ and $\overline{V}(\varphi \wedge \psi) := \overline{V}(\varphi) \cdot \overline{V}(\psi)$.

For each formula $\varphi \in \mathtt{Fmla}[\mathcal{S}]$, let X_φ be the set of propositional variables that occur in φ. Clearly, $\overline{V}(\varphi)$ depends only the values of V on X_φ. The "problem" $\mathrm{SAT}[\mathcal{S}]$ is the series $\sigma : \mathtt{Fmla}[\mathcal{S}] \longrightarrow S$ defined as follows: $\mathrm{SAT}[\mathcal{S}](\varphi) = \sum_{V \in \{0,1\}^{X_\varphi}} \overline{V}(\varphi)$.

The following corollary of our weighted version of Fagin's theorem has also appeared as [34, Thm. 6.3] with a direct proof. Our proof generalizes the reasoning for the Boolean case in [25].

▶ **Corollary 25** (Weighted Cook–Levin's theorem). *Let $\mathcal{S}$ be a finitely generated semiring. Then,* $\mathsf{SAT}[\mathcal{S}]$ *is* $\mathsf{NP}[\mathcal{S}]$*-complete.*

Now it is natural to wonder what happens with other well-known descriptive complexity results. In the reminder of this section we will tackle a few more of these. We start with the Immerman–Vardi's theorem, a result that first appeared in the Boolean case in the papers [31, 48]. Our own approach is inspired by [3, Thm. 4.4] where a version of the result for the counting complexity class FP is provided using a weighted logic with the semiring $\mathbb{N}$. We must observe, however, that our proof is a generalization of that in [3] that works for all semirings and not only $\mathbb{N}$.

▶ **Theorem 26** (Weighted Immerman–Vardi's theorem). *The logic* $\mathrm{wLFP}[\mathcal{S}]$ *(with weights in a semiring $\mathcal{S}$)* captures $\mathsf{FP}[\mathcal{S}]$ *over ordered structures in the vocabulary τ.*

▶ **Corollary 27.** *For ordered structures in a finite vocabulary τ, we have that:*

(1) $\mathrm{wLFP}[\mathbb{B}]$ captures P *(originally proved in [31, 48]).*

(2) $\mathrm{wLFP}[\mathbb{N}]$ captures FP *(originally proved in [3]).*

▶ Remark 28. Observe that using second-order Horn logic (which is known to capture P [24]) instead of least fixed-point logic, would not work for us, as in the weighted version one can encode a #P-complete problem (namely #HORNSAT). This was already noted in [3].

In the next result, $\mathrm{wPFP}[\mathcal{S}] + \{\prod X, \sum X\}$ will denote the logic that is obtained from $\mathrm{wPFP}[\mathcal{S}]$ by the addition of the second-order quantitative quantifiers $\prod X$ and $\sum X$. Clearly, when $\mathcal{S} = \mathbb{B}$, this is the same as second-order logic with partial fixed points. The Boolean counterpart of Theorem 29, namely that second-order logic extended with partial fixed points characterizes PSPACE is folklore, but a proof can be found in [41, Thm. 4]. The classical argument also uses the result for partial fixed-point logic in [1, 48] stating that the logic characterizes PSPACE over ordered structures.

▶ **Theorem 29.** *The logic* $\mathrm{wPFP}[\mathcal{S}] + \{\prod X, \sum X\}$ *(with weights in a semiring* $\mathcal{S}$*)* captures $\mathsf{FPSPACE}[\mathcal{S}]$ *over ordered structures in the vocabulary* τ*.*

▶ **Corollary 30.** *For ordered structures in a finite vocabulary* τ*, we have that:*
(1) $\mathrm{wPFP}[\mathbb{B}] + \{\prod X, \sum X\}$ captures PSPACE *(folklore, cf. [41]).*
(2) $\mathrm{wPFP}[\mathbb{N}] + \{\prod X, \sum X\}$ captures FPSPACE *(originally proved in [3]).*

▶ **Theorem 31.** *The logic* $\mathrm{wPFP}[\mathcal{S}]$ *(with weights in a semiring* $\mathcal{S}$*)* captures $\mathsf{FPSPACE}_{poly}[\mathcal{S}]$ *over ordered structures in the vocabulary* τ*.*

▶ **Corollary 32.** *For ordered structures in a finite vocabulary* τ*, we have that:*
(1) $\mathrm{wPFP}[\mathbb{B}]$ captures PSPACE *(originally proved in [1, 48]).*
(2) $\mathrm{wPFP}[\mathbb{N}]$ captures $\mathsf{FPSPACE}_{poly}$ *(originally proved in [3]).*

▶ **Theorem 33.** *The logic* $\mathrm{wDTC}[\mathcal{S}]$ *(with weights in a semiring* $\mathcal{S}$*)* captures $\mathsf{FPLOG}[\mathcal{S}]$ *over ordered structures in the vocabulary* τ*.*

▶ **Corollary 34.** *For ordered structures in a finite vocabulary* τ*, we have that:*
(1) $\mathrm{wDTC}[\mathbb{B}]$ captures DLOGSPACE *(originally proved in [31]).*
(2) $\mathrm{wDTC}[\mathbb{N}]$ captures FPLOG*.*

To end the present section, we address the general and interesting open problem suggested in [18] regarding a Fagin theorem that characterizes the class $\mathrm{NP}[\mathcal{R}]$ from [18, Def. 14]. We begin by observing that for the machine model in [18, Def. 12], Fagin's theorem will fail if the logic considered is wESO. This is essentially due to the fact that semiring Turing machines allow for arbitrary semiring values on the tape and can transition with these values. However, such a large set of transitions, is only actually needed when there are infinitely many semiring values in the input words.

▶ **Proposition 35.** *Let* $\mathcal{R}$ *be a commutative semiring. There is a series* $P \in \mathrm{NP}[\mathcal{R}]$ *such that for no* $\varphi \in \mathrm{wESO}$*,* $||\varphi|| = P$*.*

Thus one might reasonably further ask what kind of logic would capture $\mathrm{NP}[\mathcal{R}]$. Observe that an obvious challenge here is that in the proof of Fagin's theorem at some point we need to encode in the logic by means of a sentence involving a long (but finite) disjunction what the legal transitions of our machine are. Consequently, in the presence of infinitely many transitions, it is not clear how to achieve a Fagin-style characterization in a finitary language as before. Furthermore, it appears that semiring Turing machines are more suitable for an analysis that involves semirings that are not finitely generated.

By contrast to the above situation, we might ask a more restricted question if what we are doing is trying to capture $\mathrm{NP}[\mathcal{R}]$ over the class of all finite ordered structures. Recall that we are considering finite structures to be given via their binary encodings and thus the relevant series in $\mathrm{NP}[\mathcal{R}]$ are those that take as input merely binary strings. These series are not computed by SRTMs that involve infinitely many transitions because the input words do not involve semiring values. So let us consider now the modification of [18, Def. 12] that only allows semiring Turing machines to come with a finite set of transitions. In this case we will easily see that their machine model coincides with ours.

▶ **Proposition 36.** *Let $\mathcal{R}$ be a commutative semiring and allow only finitely many transitions in a semiring Turing machine. Then* $\mathrm{NP}[\mathcal{R}] = \mathsf{NP}[\mathcal{R}]$*, i.e. the NP class in the sense of [18] coincides with the NP class in our sense.*

6 Conclusions and further work

In this paper, we have established a few central results in weighted descriptive complexity, providing quantitative versions of Fagin's theorem and the Immerman–Vardi's theorem, among other logical characterizations of complexity classes. We also plan to extend our weighted Fagin's theorem to the even larger class of valuation monoids containing all semirings and supporting average calculations by the theory developed in [21] for weighted finite automata over words and weighted EMSO logic.

Furthermore, in future work, we aim to characterize further weighted complexity classes. For example, in the definition of $\mathsf{NP}[\mathcal{S}]$, by changing the requirement about polynomial time to logarithmic space on the size of the input, we can obtain a weighted complexity class that generalizes the classical counting class $\#\mathsf{L}$. The latter has been characterized by means of a logic weighted on the semiring $\mathbb{N}$ in [3, Thm. 6.4]. We suspect that this work can be generalized.

References

1 Serge Abiteboul and Victor Vianu. Fixpoint extensions of first-order logic and datalog-like languages. In *Proceedings of Fourth Annual Symposium on Logic in Computer Science*, pages 71–79. IEEE Comput. Soc. Press, 1989.

2 Jürgen Albert and Jarkko Kari. Digital image compression. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science, pages 453–479, Berlin, Heidelberg, 2009. Springer-Verlag.

3 Marcelo Arenas, Martin Muñoz, and Cristian Riveros. Descriptive complexity for counting complexity classes. *Logical Methods in Computer Science*, 16, 2020.

4 Benjamín Callejas Bedregal and Santiago Figueira. On the computing power of fuzzy turing machines. *Fuzzy Sets and Systems*, 159:1072–1083, 2008.

5 Clifford Bergman. *Universal Algebra: Fundamentals and Selected Topics*. Chapman and Hall/CRC, 2011.

6 Jean Berstel, Dominique Perrin, and Christophe Reutenauer. *Codes and Automata*. Cambridge University Press, 2009.

7 Ronald V. Book, Timothy J. Long, and Alan L. Selman. Qualitative relativizations of complexity classes. *Logical Methods in Computer Science*, 30:395–413, 1985.

8 Jin-Yi Cai and Lane A. Hemachandra. On the power of parity. In *Proceedings of the 6th Symposium on Theoretical Aspects of Computer Science*, volume 349 of *Lecture Notes in Computer Science*, pages 229–240, Berlin, 1989. Springer-Verlag.

9 Carsten Damm, Markus Holzer, and Pierre McKenzie. The complexity of tensor calculus. In *Proceedings of the 15th Annual Conference on Computational Complexity*, Lecture Notes in Computer Science, pages 70–86, Berlin, 2000. IEEE Comput. Soc. Press.

10 Carsten Damm, Markus Holzer, and Pierre McKenzie. The complexity of tensor calculus. *Computational Complexity*, 11:54–89, 2002.

11 Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.

12 Manfred Droste, Werner Kuich, and Heiko Vogler (editors). *Handbook of Weighted Automata*. Monographs in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, 2009.

13 Manfred Droste and Erik Paul. A feferman–vaught decomposition theorem for weighted mso logic. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of*

Computer Science, volume 76 of *Lecture Notes in Computer Science*, pages 1–15, Berlin, 2018. IEEE Comput. Soc. Press.

14 Arnaud Durand, Anselm Haak, Juha Kontinen, and Heribert Vollmer. Descriptive complexity of #p functions: A new perspective. *Journal of Computer and System Sciences*, 116:40–54, 2021.

15 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, Heidelberg, 1995.

16 Samuel Eilenberg. *Automata, Languages, and Machines.* Academic Press. Academic Press, New York and London, 1974.

17 Thomas Eiter and Rafael Kiesel. On the complexity of sum-of-products problems over semirings. In *Proceedings of the AAAI Conference on Artificial Intelligence AAAI-21*, pages 6304–6311. IEEE Comput. Soc. Press, 2021.

18 Thomas Eiter and Rafael Kiesel. Semiring reasoning frameworks in ai and their computational complexity. *Journal of Artificial Intelligence Research*, 77:207–293, 2023.

19 Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of computation*, 7:43–73, 1974.

20 Stephen A. Fenner, Lance J. Fortnow, and Stuart A. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48:116–148, 1994.

21 Paul Gastin and Benjamin Monmege. A unifying survey on weighted logics and weighted automata - core weighted logic: minimal and versatile specification of quantitative properties. *Soft Computing*, 22:1047–1065, 2018.

22 Christian Glasser. Space-efficient informational redundancy. *Journal of Computer and System Sciences*, 76:792–811, 2010.

23 Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the TwentySixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 31–40, Beijing, China, 2007. IEEE Comput. Soc. Press.

24 Erich Grädel. Capturing complexity classes by fragments of second-order logic. *Theoretical Computer Science*, 101:35–57, 1992.

25 Erich Grädel, Phokion Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications.* Springer-Verlag, 2007.

26 Erich Grädel and Val Tannen. Semiring provenance for first-order model checking, 2017. `arXiv:1712.01980`.

27 Erich Grädel and Val Tannen. Provenance analysis for logic and games. *Moscow Journal of Combinatorics and Number Theory*, 9:203–228, 2020.

28 S. Gupta. Closure properties and witness reduction. *Journal of Computer and System Sciences*, 50:412–432, 1995.

29 Yuri Gurevich and Saharon Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

30 Tero Harju and Juhani Karhumäki. The equivalence problem of multitape finite automata. *Theoretical Computer Science*, 78:347–355, 1991.

31 Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

32 Neil Immerman. *Descriptive Complexity.* Graduate texts in computer science. Springer, 1999.

33 Juha Kontinen. A logical characterization of the counting hierarchy. *ACM Transactions on Computational Logic*, 10:1–21, 2009.

34 Peter Kostolányi. Weighted automata and logics meet computational complexity, 2023. `arXiv:2312.10810`.

35 Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

36 Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages.* Monographs in Theoretical Computer Science. Springer Verlag, 1985.

37 Richard E. Ladner. Polynomial space counting problems. *SIAM Journal on Computing*, 18:1087–1097, 1989.

38 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Springer, 2004.

39 V. Yu. Meitus. Decidability of the equivalence problem for deterministic pushdown automata. *Cybernetics and Systems Analysis*, 5:20–45, 1992.

40 Christos H. Papadimitriou and Stathis Zachos. Two remarks on the power of counting. In *Proceedings of 6th GI Conferences in Theoretical Computer Science*, pages 269–275, 1983.

41 David Richerby. Logical characterizations of pspace. In J. Marcinkowski and A. Tarlecki, editors, *Proceedings of Computer Science Logic CSL 2004*, volume 3210 of *Lecture Notes in Computer Science*, pages 370–384, Berlin, Heidelberg, 2004. Springer.

42 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

43 Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Monographs in Computer Science. Springer, 1978.

44 Sanjeev Saluja, K. V. Subrahmanyam, and Madhukar N. Thakur. Descriptive complexity of #p functions. *Journal of Computer and System Sciences*, 50:493–505, 1995.

45 Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.

46 Géraud Senizergues. The equivalence problem for deterministic pushdown automata is decidable. In *Proceedings of International Colloquium on Automata, Languages, and Programming ICALP 1997*, volume 1256 of *Lecture Notes in Computer Science*, pages 671–681, 1997.

47 Leslie Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410–421, 1979.

48 Moshe Vardi. The complexity of relational query languages (extended abstract). In *STOC 1982 Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 137–146, 1982.

49 Jiří Wiedermann. Characterizing the super-turing computing power and efficiency of classical fuzzy turing machines,. *Theoretical Computer Science*, 317:61–69, 2004.