On the Complexity of Sum-of-Products Problems over Semirings

Thomas Eiter and Rafael Kiesel
Vienna University of Technology
Favoritenstrasse 9-11
Vienna 1040
{thomas.eiter,rafael.kiesel}@tuwien.ac.at

Abstract

Many important problems in AI, among them SAT, #SAT, and probabilistic inference, amount to Sum-of-Products Problems, i.e. evaluating a sum of products of values from some semiring R. While efficiently solvable cases are known, a systematic study of the complexity of this problem is missing. We characterize the latter by NP($\mathcal{R}$), a novel generalization of NP over semiring R, and link it to well-known complexity classes. While NP($\mathcal{R}$) is unlikely to be contained in FPSPACE(POLY) in general, for a wide range of commutative (resp. in addition idempotent) semirings, there are reductions to #P (resp. NP) and solutions are thus only mildly harder to compute. We finally discuss NP($\mathcal{R}$)-complete reasoning problems in well-known semiring formalisms, among them Semiring-based Constraint Satisfaction Problems, obtaining new insights into their computational properties.

Introduction

The Sum-of-Products Problem (SUMPROD) (Bacchus, Dalmao, and Pitassi 2009) is as follows. Given a finite domain $\mathcal{D}$ and functions $f_i : \mathcal{D}^{j_i} \to R, i = 1, \dots, n$ compute

$$\sum_{X_1,\dots,X_m \in \mathcal{D}} \prod_{i=1}^{n} f_i(\vec{Y_i}), \quad (1)$$

where $\vec{Y_i}$ is a vector of variables from $\{X_1, \dots, X_m\}$. To solve the problem, we need to compute the sum of the products of the functions f_i for all assignments of the variables X_i. The “sum” and the “product” do not need to be the usual addition and multiplication over the reals, but can be any addition $\oplus$ and multiplication $\otimes$ from a semiring $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$.

Sum-of-Products Problems, over different semirings, are present in many areas. On the one hand, many well known problems, like SAT, #SAT, MAX-SAT and Most Probable Explanation-inference are all SUMPROD-instances, when $(\oplus, \otimes)$ are $(\wedge, \vee)$, $(+, \cdot)$, $(\max, +)$ and $(\max, \cdot)$, respectively (Bacchus, Dalmao, and Pitassi 2009; Friesen and Domingos 2016). On the other hand, in many works, definitions are parameterised with semirings (Bistarelli et al. 1999; Green, Karvounarakis, and Tannen 2007; Goodman 1999; Aji and McEliece 2000; Bistarelli and Santini 2010; Eiter and Kiesel 2020; Kimmig, Van den Broeck, and De Raedt 2011). Some of them, such as Semiring-based Constraint Satisfaction Problems (SCSP) (Bistarelli et al. 1999), Algebraic Model Counting (AMC) (Kimmig, Van den Broeck, and De Raedt 2017) and Weighted First-Order Formula Evaluation (Eiter and Kiesel 2020), can be seen as instances of SUMPROD.

Due to their widespread presence in AI, solving SUMPROD-instances efficiently is of interest. Thus, compilability classes (Kimmig, Van den Broeck, and De Raedt 2017; Friesen and Domingos 2016) and fixed parameter tractability (Ganian et al. 2018) were considered, leading to efficiently solvable fragments. Apart from that, generalizations of BDDs and DPLL were considered (Wilson 2005; Dudek, Phan, and Vardi 2020; Bacchus, Dalmao, and Pitassi 2009), in order to obtain better algorithms for SUMPROD.

The computational complexity of SUMPROD, however, remained largely unconsidered. It is known that for every non-trivial, idempotent semiring the problem is NP-hard (Bistarelli et al. 1999). Furthermore, there are some results for specific semirings like the natural number semiring $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$ and the Boolean semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$. Here, SUMPROD is #P-complete and NP-complete, respectively (Stearns and Hunt III 1996). We observe that there are significant deficiencies: there is no general lower-bound and the gap between NP and #P seems very large, given that one oracle call to #P already suffices to solve any problem in the polynomial hierarchy (Toda 1989). Further, we are left without upper bounds as there are semirings that are even “harder” than $\mathbb{N}$, like the rational numbers $\mathbb{Q}$ or polynomials $\mathbb{N}[x]$. Therefore, a more detailed study of the computational complexity of SUMPROD is necessary in order to better understand the problem and the influence of the semiring parameter on its complexity.

When approaching the complexity analysis of SUMPROD over general semirings, we face a representational issue. For semirings over an infinite set R, its elements need to be *encoded* using a finite alphabet. Naturally, the time and space required to solve SUMPROD varies with different encodings (cf. Knapsack, which is polynomial for the unary encoding of the integers). In order to characterize the intrinsic complexity independently of the encoding, we introduce *Semiring Turing Machines (SRTMs)* that can have unencoded semiring values on their tape. Based on SRTMs we introduce NP($\mathcal{R}$), the class of problems solvable by SRTMs in polynomial time, as a generalization of NP over semirings that characterizes the complexity of SumProd.

To connect these results to a usual setting, we relate $\mathrm{NP}(\mathcal{R})$ to well-studied complexity classes like NP, #P, OptP for well behaved encodings of $\mathcal{R}$. We do this both for specific, common semirings but also for subclasses of commutative, idempotent, and finitely generated semirings.

We then demonstrate the power of these results by using them to derive complexity results for SumProd over a broad range of semirings and apply our theoretical results to practical problems that are parameterised with semirings.

A brief summary of our main contributions is as follows:

- We introduce Semiring Turing Machines and show that SumProd, SCSPs, AMC and weighted first-order formula evaluation over $\mathcal{R}$ are $\mathrm{NP}(\mathcal{R})$-complete for every semiring $\mathcal{R}$, where $\mathrm{NP}(\mathcal{R})$ is an analog of NP.
- We prove that for general semirings SumProd is not in FPSpace(poly), for any reasonable encoding, unless NP $\subseteq$ P/poly (which is usually assumed to be false).
- We show that for a broad subclass of commutative (resp. additionally idempotent) semirings, SumProd is counting-reducible to #P (resp. contained in $\mathrm{FP}^{\mathrm{NP}}_{\parallel}$) and derive complexity results for many specific semirings.

This work gives valuable insights into the structure of SumProd problems and shows that while there are semirings such that SumProd is unlikely to produce a polynomial output, there are many semirings for which SumProd can be efficiently reduced to SAT and #SAT.

More details (proofs etc.) will be given in a full version.

Preliminaries

Definition 1 (Semiring)**.** *A* semiring $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ *consists of a nonempty set R equipped with two binary operations $\oplus$ and $\otimes$, called addition and multiplication, where*

- $(R, \oplus)$ *is a commutative monoid with identity element* $e_\oplus$,
- $(R, \otimes)$ *is a monoid with identity element* $e_\otimes$,
- *multiplication left and right distributes over addition, and*
- $e_\oplus$ *annihilates R, i.e.* $\forall r \in R : r \otimes e_\oplus = e_\oplus = e_\oplus \otimes r$.

A semiring is commutative*, if $(R, \otimes)$ is commutative, and is* idempotent*, if* $\forall r \in R : r \oplus r = r$.

Some examples of well-known semirings are

- $\mathbb{F} = (\mathbb{F}, +, \cdot, 0, 1)$, for $\mathbb{F} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ the semiring of the numbers in $\mathbb{F}$ with addition and multiplication,
- $\mathcal{R}_{\max} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$, $\mathcal{R}_{\min} = (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$, the tropical semirings,
- $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$, the Boolean semiring,
- $\mathcal{R}[(x_i)_\alpha] = (R[(x_i)_\alpha], \oplus, \otimes, e_\oplus, e_\otimes)$, for $\alpha \in \mathbb{N}$ (resp. $\alpha = \infty$), is the semiring of polynomials with variables $x_1, \dots, x_\alpha$ (resp. $x_1, x_2, \dots$) over the semiring $\mathcal{R}$.

For our work with semirings, morphisms are important.

Definition 2 (Homomorphism, Epimorphism)**.** *Given two semirings $\mathcal{R}_i = (R_i, \oplus_i, \otimes_i, e_{\oplus_i}, e_{\otimes_i}), i = 1, 2$ a* homomorphism *(resp.* epimorphism*) from $\mathcal{R}_1$ to $\mathcal{R}_2$ is a (resp. surjective) function $f : R_1 \to R_2$ s.t. for $\odot = \oplus, \otimes$*

$$f(r \odot_1 r') = f(r) \odot_2 f(r') \text{ and } f(e_{\odot_1}) = e_{\odot_2}.$$

For our complexity considerations we need reductions. Since we are studying functional complexity, we use

Definition 3 (Metric, Counting & Karp Reduction)**.** *Let $f_i : \Sigma_i^* \to \Sigma_i^*, i = 1, 2$. A* metric reduction *from f_1 to f_2 is a pair of poly-time computable functions T_1, T_2 s.t. $T_1 : \Sigma_1^* \to \Sigma_2^*$, $T_2 : \Sigma_1^* \times \Sigma_2^* \to \Sigma_1^*$ and $f_1(x) = T_2(x, f_2(T_1(x)))$ for every $x \in \Sigma_1^*$. A metric reduction is a* counting reduction*, if $T_2(x, y) = T_2(x', y)$ for all x, x', y, and a* Karp reduction*, if $T_2(x, y) = y$ for all x, y.*

Also, we use some well-known complexity classes.

- #P (Valiant 1979) (resp. GapP (Fenner, Fortnow, and Kurtz 1994)): the functions definable as the number of accepting paths (resp. minus the number of rejecting paths) of a nondeterministic poly-time Turing Machine (NTM).
- OptP (Krentel 1988): the functions definable as the maximum output of a polynomial time NTM.
- FP (resp. FPSpace(poly)): the functions computable in poly-time (resp. poly-space with polynomial output).
- $\mathrm{FP}^{\mathrm{NP}}_{\parallel}$ (Jenner and Torán 1993): the functions computable in FP with parallel queries to an NP oracle.
- P/poly (Karp and Lipton 1982) (FP/poly): the (function) problems solvable in P (FP) with polynomial advice.

Definition 4 (Hardness, Completeness)**.** *A problem P is $\mathcal{C}$-*hard *for a complexity class $\mathcal{C}$ under X-reductions, if every problem $P' \in \mathcal{C}$ can be reduced to P by some X-reduction; P is $\mathcal{C}$-*complete *under X-reductions, if in addition $P \in \mathcal{C}$.*

Weighted Quantified Boolean Formulas

Our first goal is to characterize the functional complexity of SumProd over any semiring $\mathcal{R}$. Prior to that, we introduce weighted Quantified Boolean Formulas (QBFs) over a semiring $\mathcal{R}$ and show that the associated evaluation problem $\mathrm{SAT}(\mathcal{R})$ is equivalent to SumProd over $\mathcal{R}$. This makes explicit that SumProd-instances are not necessarily pure calculations but are influenced by qualitative conditions. Furthermore, it highlights how SAT, #SAT and other well-known problems fit into the context of SumProd.

We define weighted QBFs similarly to other weighted logics (Droste and Gastin 2007; Mandrali and Rahonis 2015).

Definition 5 (Syntax)**.** *Let $\mathcal{V}$ be a set of propositional variables and $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ be a semiring. A* weighted QBF over $\mathcal{R}$ *is of the form α given by*

$$\alpha ::= k \mid v \mid \neg v \mid \alpha + \alpha \mid \alpha * \alpha \mid \Sigma v \alpha \mid \Pi v \alpha$$

where $k \in R$ and $v \in \mathcal{V}$. A weighted fully quantified Boolean Formula *is a weighted QBF without free variables.*

Definition 6 (Semantics)**.** *A subset $\mathcal{I}$ of $\mathcal{V} \cup \{\neg v \mid v \in \mathcal{V}\}$ is an* interpretation *if $\neg v \in \mathcal{I} \Leftrightarrow v \notin \mathcal{I}$ for every $v \in \mathcal{V}$. Given a weighted QBF α over a semiring $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ and an interpretation $\mathcal{I}$, the semantics $[\![\alpha]\!]_\mathcal{R}(\mathcal{I})$ of α over $\mathcal{R}$ w.r.t. $\mathcal{I}$ is defined as*

$$[\![k]\!]_\mathcal{R}(\mathcal{I}) = k$$

$$[\![l]\!]_\mathcal{R}(\mathcal{I}) = \begin{cases} e_\otimes & l \in \mathcal{I} \\ e_\oplus & \textit{otherwise.} \end{cases} \quad (l \in \{v, \neg v\})$$

$$\llbracket \alpha_1 + \alpha_2 \rrbracket_{\mathcal{R}}(\mathcal{I}) = \llbracket \alpha_1 \rrbracket_{\mathcal{R}}(\mathcal{I}) \oplus \llbracket \alpha_2 \rrbracket_{\mathcal{R}}(\mathcal{I})$$
$$\llbracket \alpha_1 * \alpha_2 \rrbracket_{\mathcal{R}}(\mathcal{I}) = \llbracket \alpha_1 \rrbracket_{\mathcal{R}}(\mathcal{I}) \otimes \llbracket \alpha_2 \rrbracket_{\mathcal{R}}(\mathcal{I})$$
$$\llbracket \Sigma v \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}) = \llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}_v) \oplus \llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}_{\neg v})$$
$$\llbracket \Pi v \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}) = \llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}_v) \otimes \llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I}_{\neg v})$$

where $\mathcal{I}_v = \mathcal{I} \setminus \{\neg v\} \cup \{v\}$ *and* $\mathcal{I}_{\neg v} = \mathcal{I} \setminus \{v\} \cup \{\neg v\}$.

Weighted QBFs generalize QBFs in negation normal form, as negation is only allowed in front of variables. Here, we further focus on ΣBFs, i.e., the weighted fully quantified BFs that contain only sum quantifiers (i.e. only Σv) to fit into the context of SUMPROD. As for evaluation, we introduce:

SAT($\mathcal{R}$): given a ΣBF α over $\mathcal{R}$ compute $\llbracket \alpha \rrbracket_{\mathcal{R}}(\emptyset)$.

Example 1. *Over $\mathbb{B}$, the Boolean semiring,* SAT($\mathbb{B}$) *is* SAT, *as $0, 1, +, *, \Sigma v$ are $\bot, \top, \vee, \wedge, \exists v$, respectively.*

Further, LEXMAXSAT, *the problem of obtaining the lexicographically maximum satisfying assignment, for a propositional formula ϕ can be expressed over the semiring $\mathcal{R}_{\max}$ using the ΣBF $\Sigma v_1 \dots \Sigma v_n \phi * \Pi_{i=1}^{n}(v_i * 2^{n-i} + \neg v_i)$.*

When the functions f_i in the expression (1) are explicit maps from variable assignments to semiring values (represented using some finite alphabet), we obtain:

Theorem 7. SAT($\mathcal{R}$) *is Karp-reducible to* SUMPROD *over semiring $\mathcal{R}$ and vice versa for every semiring $\mathcal{R}$.*

In the Boolean case, this follows simply from the fact that 3SAT is NP-complete. However, for semirings in general it is more difficult, since the Tseitin-transformation (Tseitin 1983), which is used for this proof, cannot be lifted. Instead, we need to implicitly exploit the distributive law during evaluation as in Algorithm 1.

Example 2. *Consider the term $5 + 7 * (3 + 9)$, which is not a sum of products. Algorithm 1 would nondeterministically return an element from $\{5, 7 * 3, 7 * 9\}$.*

Evidently, SAT($\mathcal{R}$) is similar to #P and OPTP, where values are nondeterministically generated and aggregated:

Proposition 8. *The sum, using $\oplus$, of the results of all execution paths for a call to* EVAL$_{\mathcal{R}}(\alpha, \mathcal{I})$ *is equal to $\llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I})$.*

Algorithm 1 An algorithm for weighted formula evaluation.

Input A ΣBF α over semiring $\mathcal{R}$ and an interpretation $\mathcal{I}$.
Output Nondeterministically, all summands of $\llbracket \alpha \rrbracket_{\mathcal{R}}(\mathcal{I})$.

function EVAL$_{\mathcal{R}}(\alpha, \mathcal{I})$
  switch α **do**
    case $\alpha = k$: **return** k
    case $\alpha = l, l \in \{v, \neg v\}$:
      if $l \in \mathcal{I}$ **then**: **return** $e_{\otimes}$
      else: **return** $e_{\oplus}$
    case $\alpha = \alpha_1 + \alpha_2$:
      Guess $i \in \{1, 2\}$
      return EVAL$_{\mathcal{R}}(\alpha_i, \mathcal{I})$
    case $\alpha = \alpha_1 * \alpha_2$:
      return EVAL$_{\mathcal{R}}(\alpha_1, \mathcal{I}) \otimes$ EVAL$_{\mathcal{R}}(\alpha_2, \mathcal{I})$
    case $\alpha = \Sigma v \alpha$:
      Guess $\mathcal{I}' \in \{\mathcal{I} \setminus \{\neg v\} \cup \{v\}, \mathcal{I} \setminus \{v\} \cup \{\neg v\}\}$
      return EVAL$_{\mathcal{R}}(\alpha, \mathcal{I}')$

While this computation can be simulated in #P, GAPP and OPTP when $\mathcal{R}$ is $\mathbb{N}, \mathbb{Z}, \mathcal{R}_{\max}$, respectively, we are not aware of such possibilities in general. Thus, a linkage is difficult. Our plan is therefore as follows. We introduce a new abstract model of computations over semirings, which we then connect to well-known complexity classes.

Semiring Turing Machines

We generalize NTMs to Semiring Turing Machines (SRTMs) to characterize the complexity of SUMPROD. The latter should thus be capable of

- semiring operations irrespective of encodings of values;
- summing up values generated by nondeterministic computations; and
- using input-values in calculations.

On the other hand, too much power should be avoided; to this end, we relegate computation to weighted transitions.

Definition 9 (SRTM)**.** *A* Semiring *Turing Machine is a 7-tuple $M = (\mathcal{R}, R', Q, \Sigma, \iota, \sqcup, \delta)$, where*

- *$\mathcal{R}$ is a semiring*
- *$R' \subseteq R$ is a finite set of semiring values*
- *Q is a finite set of states*
- *Σ is a finite set of symbols (the tape alphabet)*
- *$\iota \in Q$ is the initial state*
- *$\sqcup \in \Sigma$ is the blank symbol*
- *$\delta \subseteq (Q \times (\Sigma \cup R)) \times (Q \times (\Sigma \cup R)) \times \{-1, 1\} \times R$ is a weighted transition relation, where the last entry of the tuple is the weight. For each $((q_1, \sigma_1), (q_2, \sigma_2), e, r) \in \delta$:*
  1. *$\sigma_1 \in R$ or $\sigma_2 \in R$ implies $\sigma_1 = \sigma_2$ (cannot write or overwrite semiring values),*
  2. *$r \in R'$ or $r = \sigma_1 \in R$ (transition only with $r \in R'$ or value under head), and*
  3. *$\sigma_1 \in R$ implies that for all $\sigma_1' \in R$ we have $((q_1, \sigma_1'), (q_2, \sigma_1'), e, r') \in \delta$, where $r' = \sigma_1'$ if $r = \sigma_1$ and $r' = r$ otherwise (cannot differentiate semiring values).*

As usual, -1 and 1 move the head to the left and right. The output of a computation is as follows.

Definition 10 (SRTM function)**.** *The* value $v(c)$ *of an SRTM M on a configuration $c = (q, x, n)$, where q is a state, x is the string on the tape and n is the head position, is recursively defined by $v(c) = \bigoplus_{c \xrightarrow{r} c'} r \otimes v(c')$, where $c \xrightarrow{r} c'$ denotes that M can transit from c to c' with weight r; the empty sum has value $e_{\otimes}$. The* output *of M on input x is $v(\iota, x, 0)$.*

With this in place, we define an analog of NP.

Definition 11 (NP($\mathcal{R}$))**.** NP($\mathcal{R}$) *is the class of all functions computable in polynomial time by an SRTM over $\mathcal{R}$.*

Intuitively, SRTMs work similarly to the functions definable in OPTP. We are allowed to nondeterministically generate output values, which are then aggregated. For SRTMs the aggregation function is the sum $\oplus$ of the semiring and for OPTP it is max. However, contrary to OPTP-functions, SRTMs cannot generate outputs using arbitrary manipulations due to the restrictions on the transition function. Instead, they always generate a product of the semiring values from the input and a finite set R', which is fixed before execution. It would be possible to allow non-recursively defined sums and products of semiring values on the tape. However, we decided against this option as we can simulate such more complicated features and thus avoid more "semantic" restrictions that require program analysis to be verified.

Furthermore, SRTMs cannot make decisions based on the semiring values in the input but must treat them as black boxes. Otherwise, SRTMs could for example decide whether $n \in \mathbb{N}$ is prime in constant time. This is why we need condition 3. on the transition relation, which also implies that while the transition relation may not be finite, it can always be finitely represented.

SRTMs are well suited to characterize the complexity of sum-of-products problems:

Theorem 12. $\mathrm{SAT}(\mathcal{R})$ *is* $\mathrm{NP}(\mathcal{R})$*-complete w.r.t. Karp reductions for every semiring* $\mathcal{R}$*.*

Proof (sketch). Membership can be seen easily from Algorithm 1. All that is needed is to change "return k" to "transition, with the weight currently on the tape" and return $e_\oplus/e_\otimes$ to "transition into the next state with weight $e_\oplus/e_\otimes$", respectively. To prove the hardness, we can generalize the Cook-Levin Theorem, cf. (Gary and Johnson 1979). □

Corollary 13. SUMPROD *over* $\mathcal{R}$ *is* $\mathrm{NP}(\mathcal{R})$*-complete w.r.t. Karp reductions for every semiring* $\mathcal{R}$*.*

Relation to Known Complexity Classes

These results characterize the complexities of sum-of-products problems over different semirings, independently of encodings. In order to gain more insight into their complexity in a usual setting, we relate $\mathrm{NP}(\mathcal{R})$ to well-known complexity classes.

For this purpose, we must encode semiring values in a finite tape alphabet and thus introduce the following notions.

Definition 14 (Encoding Function, Encoded Semiring)**.** *Let* $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ *be a semiring. Then an injective function* $e : R \to \{0,1\}^*$ *is an* encoding function*.*

We let $e(\mathcal{R}) = (e(R), \oplus, \otimes, e(e_\oplus), e(e_\otimes))$ *denote the* encoded semiring *given by* $e(R) = \{e(r) \mid r \in R\}$ *and* $\odot$ *on* $e(R)$*, s.t.* $e(r_1) \odot e(r_2) = e(r_1 \odot r_2)$ *for* $\odot = \oplus, \otimes$*.*

Given an encoded value $e(r)$ *we define* $\|r\|_e$*, the* size of r w.r.t. e*, as the length of the bitstring* $e(r)$*, i.e.* $|e(r)|$*.*

Now, we can use classical machines to solve $\mathrm{SAT}(e(\mathcal{R}))$ and consider the complexity of the problem. It depends on the complexity of addition and multiplication. While for $\mathbb{N}, \mathbb{B}$ these operations are "easy", i.e., feasible in polynomial time given a binary encoding, this is not the case for arbitrary semirings. A single multiplication may even be undecidable.

Example 3 (Arbitrary Complexity Semirings)**.** *Given* $M \subseteq \{0,1\}^*$ *and* $\succ$*, the lexicographical order on* $\{0,1\}^*$*, we define the semiring* $\mathcal{R}_M = (\{0,1\}^* \cup \{\mathbf{0}, \mathbf{1}\}, \max_\succ, \otimes, \mathbf{0}, \mathbf{1})$*, where* $\mathbf{1} \succ m \succ \mathbf{0}$ *for* $m \in \{0,1\}^*$ *and*

$$m_1 \otimes m_2 := \begin{cases} \min_\succ(m_1, m_2) & m_1, m_2 \in M \cup \{\mathbf{0}\} = S \\ m_1 & m_1 \in S, m_2 \notin S \\ m_2 & m_2 \in S, m_1 \notin S \\ \min_\succ(m_1, m_2) & \textit{otherwise.} \end{cases}$$

Then multiplication requires deciding $m_i \in M$*. When* M *corresponds to the halting problem, we have undecidability.*

However, the difficulty stems from the encoding.

Example 4 (cont.)**.** *If the encoding* e *maps* $m \in \{0,1\}^*$ *to* $(m, 1)$ *if* $m \in M$ *and to* $(m, 0)$ *if* $m \notin M$*, then multiplication and addition in* $e(\mathcal{R}_M)$ *are computable in linear time.*

Our intuition is that there are two sources of complexity. One seems to be the encoding and the other the amount of information that the weighted semantics gives us about the formula. The latter is determined by the non-collapsing terms in the semiring. E.g. over $\mathbb{N}[(x_i)_k]$ the coefficients c_1, c_2 are retained by the sum $c_1x_1 + c_2x_2$, over $\mathbb{N}$ only the sum $c_1 + c_2$ is retained after addition, and over $\mathbb{B}$ the value $c_1 \vee c_2$ only tells us if at least one of the values was 1. As a consequence $\mathrm{SAT}(\mathbb{N})$-instances seem to be strictly harder than $\mathrm{SAT}(\mathbb{B})$-instances as $\mathrm{NP} \subseteq \mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ (Toda 1989).

We focus on the second source of complexity and address the first by introducing the notion of an efficient encoding.

Definition 15 (Efficiently Encoded Semiring)**.** *Let* $e(\mathcal{R})$ *be an encoded semiring. Then* $e(\mathcal{R})$ *is* efficiently encoded*, if there exists a polynomial* $p(x)$ *s.t. for all* $e(r_1), \ldots, e(r_n) \in e(R)$ *it holds that*

1. $\|\bigotimes_{i=1}^n r_i\|_e \leq p(n) \max_{i=1,\ldots,n} \|r_i\|_e$*,*
2. $\|\bigoplus_{i=1}^n r_i\|_e \leq p(\log_2(n)) \max_{i=1,\ldots,n} \|r_i\|_e$*, and*
3. $e(r), e(r') \mapsto e(r \odot r')$ *is in* FP *for* $\odot = \oplus, \otimes$*.*

Conditions 1) and 2) ensure that successive multiplications resp. additions do not cause space explosion, even for sums with exponential size n. Condition 3) is obvious.

Example 5. *With binary representation* $\mathrm{bin}(n) = b_0 \ldots b_m$ *s.t.* $n = \sum_{i=1}^m b_i 2^i$*, the semiring* $\mathbb{N}$ *of the natural numbers is efficiently encoded. However,* $\mathbb{N}[(x_i)_\infty]$ *is not efficiently encoded when for a polynomial* $\sum_{\vec{i} \in I} c_{\vec{i}} x_{i_1}^{e_{i_1}} \ldots x_{i_n}^{e_{i_n}}$ *the coefficients* $c_{\vec{i}}$ *are in binary while the exponents* e_{i_j} *and variable indices* i_j *are in unary representation.*

The conditions of Definition 15 are mild in practice, as besides $\mathbb{N}$ many common semirings, e.g. $\mathbb{Z}, \mathbb{Q}, \mathcal{R}_{\max}$, are efficiently encodable. They remain so under sharpenings like $p(n) = O(n)$, but this may lead to less "natural" encodings. Efficient encodings enable space-efficient ΣBF evaluation.

Proposition 16. *If* $e(\mathcal{R})$ *is an efficiently encoded semiring, then* $\mathrm{SAT}(e(\mathcal{R}))$ *is in* FPSPACE(POLY)*.*

Results for Specific Semirings

We relate classical complexity classes to $\mathrm{NP}(e(\mathcal{R}))$ by showing that they share $\mathrm{SAT}(e(\mathcal{R}))$ as a complete problem for different specific semirings.

Theorem 17. *For* $(\mathcal{R}, \mathcal{C}) = (\mathbb{B}, \mathrm{NP}), (\mathbb{N}, \#\mathrm{P}), (\mathbb{Z}, \mathrm{GapP}), (\mathcal{R}_{\max}, \mathrm{OptP})$ *and the binary representation* bin *of the integers,* $\mathrm{SAT}(\mathrm{bin}(\mathcal{R}))$ *is* $\mathcal{C}$*-complete w.r.t. Karp reductions.*

Proof (sketch). Membership holds as $\mathrm{bin}(n)$ satisfies Definition 15. For hardness we use reductions from SAT, #SAT, computing the permanent of an integer matrix, and LEXMAXSAT, respectively. □

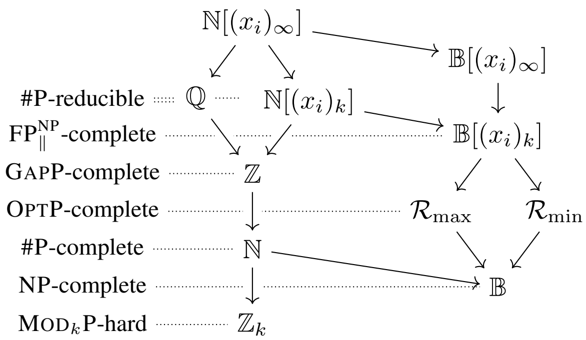

Figure 1: Epimorphisms $f : \mathcal{R}_1 \to \mathcal{R}_2$ between semirings, indicated by arrows $\mathcal{R}_1 \to \mathcal{R}_2$. Relation of complexity classes $\mathcal{C}$ and semirings $\mathcal{R}$, indicated by dotted lines $\mathcal{C} \cdots \mathcal{R}$.

Note that there are functions in OptP that cannot be computed in $\text{NP}(\text{bin}(\mathcal{R}_{\max}))$, e.g., $x \mapsto 2^{|x|}$. Informally, SRTMs can only generate semiring values by multiplying numbers from a finite set R' or the input. Given that multiplication in $\mathcal{R}_{\max}$ is $+$, we can only generate numbers that are polynomial in the numbers in the input and R'. For NP,#P and GapP this effect does not occur.

Results for Classes of Semirings

Apart from completeness results for specific semirings, we also care about intuition on why some semirings come with a higher complexity than others, and about results that help to characterize new semirings based on their properties. Thus, we consider the complexity of classes of semirings.

Our strategy is the following. We characterize the complexity over semirings, whose weighted semantics preserves the most information and that are therefore the "hardest" in a class of semirings and derive an upper bound for the whole class. Formally, we employ the following theorem.

Theorem 18. *Let $e_i(\mathcal{R}_i), i = 1, 2$ be two encoded semirings, such that*

1. $\text{SAT}(e_1(\mathcal{R}_1))$ *is in* FPSpace(poly)*,*
2. *there exists a polynomial time computable epimorphism $f : e_1(R_1) \to e_2(R_2)$, and*
3. *for each $e_2(r_2) \in e(R_2)$ one can compute in polynomial time $e_1(r_1)$ s.t. $f(e_1(r_1)) = e_2(r_2)$ from $e_2(r_2)$.*

Then $\text{SAT}(e_2(\mathcal{R}_2))$ *is counting-reducible to* $\text{SAT}(e_1(\mathcal{R}_1))$*.*

Figure 1 depicts between which semirings we can hope to apply the above theorem. $\mathbb{N}[(x_i)_\infty]$ and $\mathbb{B}[(x_i)_\infty]$ are the most information preserving and therefore "hardest" commutative and idempotent countable semirings, respectively. However, we obtain negative results for condition 1.

Theorem 19. *Let $\mathcal{R} = \mathbb{N}[(x_i)_\infty]$ (resp. $\mathcal{R} = \mathbb{B}[(x_i)_\infty]$). If there is an encoding function e for $\mathcal{R}$ s.t.*

1) *$\|[\![\alpha]\!]_\mathcal{R}(\mathcal{I})\|_e$ is polynomial in the size of α, $\mathcal{I}$,*
2) *we can extract the coefficient of $x_{i_1}^{j_1} ... x_{i_n}^{j_n}$ from $e(r)$ in polynomial time in $\|r\|_e$, and*
3) *$\|x_i\|_e$ is polynomial in i,*

then #P $\subseteq$ FP/poly *(resp.* NP $\subseteq$ P/poly*).*

This would imply $\Sigma_2^p = \text{PH}$ (Karp and Lipton 1982), which is considered to be unlikely. Hence, for any reasonable encoding e, $\text{SAT}(e(\mathbb{N}[(x_i)_\infty])$ and $\text{SAT}(e(\mathbb{B}[(x_i)_\infty])$ are unlikely to have polynomial output and, therefore, are unlikely to be in FPSpace(poly).

Condition 3) of Theorem 19 allows that indices i_k are encoded in unary and imposes no restriction on the encoding of exponents j_k. Requiring it to be in binary puts $\text{SAT}(e(\mathbb{B}[x]))$ outside of FPSpace(poly), unless NP $\subseteq$ P/poly.

Theorem 20. *Let $\mathcal{R} = \mathbb{N}[x]$ (resp. $\mathcal{R} = \mathbb{B}[x]$). If there is an encoding function e for $\mathcal{R}$ s.t.*

1) *$\|[\![\alpha]\!]_\mathcal{R}(\mathcal{I})\|_e$ is polynomial in the size of α, $\mathcal{I}$,*
2) *we can extract the coefficient of x^i from $e(r)$ in polynomial time in $\|r\|_e$, and*
3) *$\|x^i\|_e$ is polynomial in $\log_2(i)$,*

then #P $\subseteq$ FP/poly *(resp.* NP $\subseteq$ P/poly*).*

Proof (sketches) for Theorems 19 and 20. For each $n \in \mathbb{N}$, we can construct a ΣBF formula α of polynomial size s.t. the solution of every #SAT (resp. SAT)-instance with n variables is obtainable from $\|[\![\alpha]\!]_\mathcal{R}(\mathcal{I})\|_e$. By the methodology of Cadoli, Donini, and Schaerf (1996) to assess compilability, we then infer #P $\subseteq$ FP/poly (resp. NP $\subseteq$ P/poly). ☐

Notably, if #P $\subseteq$ FP/poly (resp. NP $\subseteq$ P/poly) holds, then an encoding can be given that satisfies conditions 1) - 3) of Theorem 19 (resp. 20). Thus, the existence of an encoding e satisfying conditions 1)-3) is *equivalent* to the open problem whether #P $\subseteq$ FP/poly (resp. NP $\subseteq$ P/poly) is true.

We see that reducing $\text{SAT}(e(\mathcal{R}))$ to the polynomial semirings is not practical for encoded semirings $e(\mathcal{R})$ in general. However, we obtain positive results when restricting ourselves to polynomials with a fixed number of variables. We allow rational coefficients instead of natural numbers to obtain a stronger result.

Theorem 21. *Let e be the encoding function that represents exponents in unary and coefficients in binary. Then*

- $\text{SAT}(e(\mathbb{Q}[(x_i)_k]))$ *is counting-reducible to* #SAT *and* #P*-hard for counting reductions.*
- $\text{SAT}(e(\mathbb{B}[(x_i)_k]))$ *is* $\text{FP}_{\parallel}^{\text{NP}}$*-complete for metric reductions.*

As an immediate consequence, we see that $\text{SAT}(e(\mathbb{Q}[(x_i)_k]))$ and $\text{SAT}(e(\mathbb{B}[(x_i)_k]))$ are not significantly harder than #P and NP, respectively. We can use this to obtain containment results for finitely generated commutative (idempotent) semirings.

Definition 22 ((Finitely) Generated Semiring)**.** *Let $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ be a semiring. For any $R^\star \subseteq R$, the semiring generated by $R^\star$, denoted $\langle R^\star \rangle_\mathcal{R}$, is the least (w.r.t. $\subseteq$) semiring $(R', \oplus, \otimes, e_\oplus, e_\otimes)$ s.t. $R^\star \subseteq R'$. We call $\mathcal{R}$* finitely generated*, if $\mathcal{R} = \langle R^\star \rangle_\mathcal{R}$ for some finite $R^\star$.*

Semirings that are finitely generated using k elements $r_1, \dots r_k$ can be seen as reduced versions of polynomials with variables $x_1, \dots, x_k$. We thus obtain:

Theorem 23. *Let $e(\mathcal{R})$ be an efficiently encoded commutative semiring that is generated by $\{r_1, \dots, r_k\}$. Suppose every $r \in R$ is of the form $r = \bigoplus_{i=1}^{n} \bigoplus_{l=1}^{a_i} \bigotimes_{j=i}^{m_i} r_j^{e_{i,j}}$ for some $a_i, e_{i,j}, m_i \in \mathbb{N}$ such that*

- $\max\{e_{i,j}, \log_2(a_i)\}$ *is polynomial in* $\|r\|_e$*, and*
- *we can obtain* $a_i, e_{i,j}$ *from* $e(r)$ *in polynomial time.*

Then $\mathrm{SAT}(e(\mathcal{R}))$ *is counting-reducible to* #SAT. *If* $e(\mathcal{R})$ *is in addition idempotent, then* $\mathrm{SAT}(e(\mathcal{R}))$ *is in* $\mathrm{FP}^{\mathrm{NP}}_{\|}$.

Note that if $e(\mathcal{R})$ is idempotent, we may w.l.o.g. assume $a_i = 1$. The proof of Theorem 23 uses Theorems 18 and 21.

Derived Results

As a consequence of Theorems 18 and 21, or directly from Theorem 23 for the finitely generated $\mathbb{N}, \mathbb{Z}$, we obtain:

Theorem 24. *Let* $\mathbb{S} = \mathbb{N}, \mathbb{Z}, \mathbb{Q}$. *For* $\mathbb{S}^n$*, the semiring* $\mathbb{S}$ *over multiple dimensions and* $\mathbb{S}^{n\times n}$*, the semiring of matrices with entries in* $\mathbb{S}$*, we have that* $\mathrm{SAT}(\mathbb{S}^n)$ *and* $\mathrm{SAT}(\mathbb{S}^{n\times n})$ *are counting-reducible to* #SAT.

As an example, we consider in more detail the semiring $\mathrm{GRAD} = (\mathbb{Q}_{\geq 0} \times \mathbb{Q}, +, \otimes, (0,0), (1,0))$, where addition is coordinate-wise and $(a_1, b_1) \otimes (a_2, b_2) = (a_1 \cdot a_2, a_2 \cdot b_1 + a_1 \cdot b_2)$. It was introduced by Eisner (2002) and shown to be useful for parameter optimization (Kimmig, Van den Broeck, and De Raedt 2017; Manhaeve et al. 2018).

GRAD is finitely generated by $\{(1,0), (0,1)\}$. This means that we can see the elements in GRAD as elements in $\mathbb{N}[x]$ by identifying $(1,0)$ and $(0,1)$ with 1 and x, respectively. The elements in GRAD are however only reduced versions of the polynomials, i.e., there are additional equalities between values in GRAD that do not hold in $\mathbb{N}[x]$. An example is x^2, because $(0,1) \otimes (0,1) = (0,0)$ but $x^2 \neq 0$.

Corollary 25. *By Theorems 18 and 23,* SAT(GRAD) *is counting-reducible to* #SAT *and* #P*-hard w.r.t. counting reductions, respectively.*

Applications of the Results

We now apply the results from above to problems in AI.

Weighted First-Order Logic

Weighted first-order logics were introduced by Mandrali and Rahonis (2015) for expressivity characterizations and by Eiter and Kiesel (2020) for a quantitative extension of ASP.

They are defined over a *signature* $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X} \rangle$ with predicates $p \in \mathcal{P}$ that have a fixed arity $r(p) \in \mathbb{N}$ over a domain $\mathcal{D}$ and variables in $\mathcal{X}$.

Definition 26 (Syntax)**.** *Let* $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X} \rangle$ *be a signature and* $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ *be a semiring. The* weighted σ-formulas *over* $\mathcal{R}$ *are of the form* α *given by*

$$\alpha ::= k \mid p(\vec{x}) \mid \neg p(\vec{x}) \mid \alpha + \alpha \mid \alpha * \alpha \mid \Sigma x \alpha \mid \Pi x \alpha.$$

Here $k \in R$, $p \in \mathcal{P}$, $\vec{x} \in (\mathcal{D} \cup \mathcal{X})^{r(p)}$ *and* $x \in \mathcal{X}$. *A* weighted σ-sentence *is a weighted* σ*-formula that does not contain free variables.*

Note that we again only allow negation in front of $p(\vec{x})$.

Definition 27 (Semantics)**.** *A* σ-interpretation *is a subset* $\mathcal{I}$ *of* $\{p(\vec{x}), \neg p(\vec{x}) \mid p \in \mathcal{P}, \vec{x} \in \mathcal{D}^{r(p)}\}$ *s.t.* $\neg p(\vec{x}) \in \mathcal{I} \Leftrightarrow p(\vec{x}) \notin \mathcal{I}$ *for all* $p \in \mathcal{P}, \vec{x} \in \mathcal{D}^{r(p)}$. *Given a weighted* σ*-sentence* β *and a* σ*-interpretation* $\mathcal{I}$ *the semantics* $[\![\beta]\!]^\sigma_\mathcal{R}(\mathcal{I})$ *of* β *over* $\mathcal{R}$ *w.r.t.* $\mathcal{I}$ *is defined as*

$$[\![\Sigma x \beta]\!]^\sigma_\mathcal{R}(\mathcal{I}) = \textstyle\bigoplus_{d \in \mathcal{D}} [\![\beta\{x \mapsto d\}]\!]^\sigma_\mathcal{R}(\mathcal{I})$$
$$[\![\Pi x \beta]\!]^\sigma_\mathcal{R}(\mathcal{I}) = \textstyle\bigotimes_{d \in \mathcal{D}} [\![\beta\{x \mapsto d\}]\!]^\sigma_\mathcal{R}(\mathcal{I})$$

The rest of the cases are as in Definition 6, where we identify $p(\vec{x})$ *with a propositional variable* $v_{p(\vec{x})}$.

We consider the evaluation of ΣFO σ-formulas, which only use sum quantifiers (i.e. Σx):

ΣFO-EVAL($\mathcal{R}$): Given a weighted ΣFO σ-sentence α over the semiring $\mathcal{R}$ and a σ-interpretation $\mathcal{I}$, compute $[\![\alpha]\!]^\sigma_\mathcal{R}(\mathcal{I})$.

Probabilistic inference in Bayesian networks corresponds to ΣFO-EVAL($\mathbb{N}$) (Van den Broeck, Meert, and Darwiche 2014); further, Eiter and Kiesel (2020) showed that aggregation and other extensions of Answer Set Programming can be modeled as ΣFO-EVAL($\mathcal{R}$) over different semirings $\mathcal{R}$.

The problem is very similar to SAT($\mathcal{R}$). Indeed, under the assumption that $\mathcal{I}$ is given as a bitmap, we obtain:

Theorem 28. *Problem* ΣFO-EVAL($\mathcal{R}$) *is Karp-reducible to* SAT($\mathcal{R}$) *and vice versa, and thus* NP($\mathcal{R}$)*-complete w.r.t. Karp reductions for every semiring* $\mathcal{R}$.

Proof (sketch). $\Rightarrow$: Let α be a ΣBF over $\mathcal{R}$. We choose $\sigma = \langle \{\bot, \top\}, \{t(.)\}, \{x_{v_1}, \dots, x_{v_n}\} \rangle$ and $\mathcal{I} = \{t(\top), \neg t(\bot)\}$ and replace every propositional variable v in α by $t(x_v)$.
$\Leftarrow$: We replace $p(\vec{x})$ by $\Sigma_{p(\vec{d}) \in \mathcal{I}} \Pi_{x_i \in \vec{x}} v_{x_i, d_i}$, where $\vec{x} = x_1, \dots, x_{r(p)}$, $\vec{d} = d_1, \dots, d_{r(p)}$, and $v_{x,d}$ means that variable x has value d. We add constraints s.t. when both v_{x_i,d_i} and v_{x_i,d'_i} are true, then $d_i = d'_i$ and add a quantifier $\Sigma v_{x_i,d_i}$ for each pair (x_i, d_i). ☐

Semiring-based Constraint Satisfaction Problems

Bistarelli et al. (1999) introduced a generalization of constraint satisfaction problems parameterised with *c-semirings* $\mathcal{R}$, which are idempotent commutative semirings such that the axiom $\forall r \in R : r \oplus e_\otimes = e_\otimes$ holds.

Definition 29 (Constraint System, Constraint Problem)**.** *A* constraint system *is a tuple* $CS = \langle \mathcal{R}, D, V \rangle$*, where* $\mathcal{R}$ *is a c-semiring,* D *is a finite domain, and* V *is an ordered set of variables. A constraint over* CS *is a pair* $\langle def, con \rangle$*, where* $con \subseteq V$*, and* $def : D^{con} \to R$ *is the* value *of the constraint.*

A constraint problem P *over* CS *is a pair* $P = \langle C, con \rangle$*, where* C *is a multiset of constraints over* CS *and* $con \subseteq V$.

SCSPs correspond to classical CSP, probabilistic CSP, weighted CSP and fuzzy CSP when the chosen c-semiring is $\mathbb{B}$, $([0,1], \max, \cdot, 0, 1)$, $\mathcal{R}_{\max}$ and $([0,1], \max, \min, 0, 1)$, respectively (Bistarelli et al. 1999). The two main operations on constraints are combination $*$ and projection $\Downarrow$.

Definition 30 (Combination, Projection)**.** *The* combination $c_1 * c_2$ *of two constraints* $c_i = \langle def_i, con_i \rangle, i = 1, 2$ *is the constraint* $c = \langle def_1 \otimes def_2, con_1 \cup con_2 \rangle$. *The* projection $c \Downarrow_{con'}$ *of a constraint* $c = \langle def, con \rangle$ *to* $con' \subseteq con$ *is* $\langle def', con' \rangle$ *with* $def'(t') = \bigoplus_{\{t \mid t \downarrow^{con}_{con'} = t'\}} def(t)$.

Intuitively, combination $*$ is the product ($\otimes$) of constraint values and projection $\Downarrow_{con'}$ is the sum ($\oplus$) over all assignments to the variables in $con \setminus con'$. Using $*, \Downarrow$, the consistency-level of an SCSP is defined as follows.

Definition 31 (Consistency-Level)**.** *Given an SCSP problem $P = \langle C, con \rangle$, we define the* best level of consistency *of P as $blevel(P) = (\Pi_{c \in C} c) \Downarrow_{\emptyset}$.*

We see that the computation of $blevel(P)$ is a sum of the products of the values of the constraints in P over all possible variable assignments. If def is given as a map of variable assignments to semiring values, we obtain:

Theorem 32. *Computing $blevel(.)$ over $\mathcal{R}$ is Karp-reducible to* SUMPROD *over $\mathcal{R}$ and vice versa, and thus* NP($\mathcal{R}$)*-complete w.r.t. Karp reductions for every semiring $\mathcal{R}$.*

Proof (sketch). The variables of the constraint problem correspond to the ones of SUMPROD and for each constraint $\langle def_i, con_i \rangle$ the function def_i corresponds to a function f_i in SUMPROD; $blevel(.)$ is the solution of SUMPROD. □

Algebraic Model Counting

Algebraic Model Counting (AMC) was introduced by Kimmig, Van den Broeck, and De Raedt (2017) as a generalization of weighted model counting.

Definition 33 (AMC)**.** *Given a propositional theory T over variables V, a commutative semiring $\mathcal{R}$, and a labeling function $\alpha : L \to R$ that maps the literals L over V to R,* AMC *is to compute the value*

$$A(T) = \bigoplus_{\mathcal{I} \subseteq V, \, s.t. \, \mathcal{I} \models T} \bigotimes_{v \in \mathcal{I}} \alpha(v) \otimes \bigotimes_{v \notin \mathcal{I}} \alpha(\neg v).$$

Besides the standard applications in SAT, #SAT and probabilistic inference, the authors showed that AMC can be used to perform sensitivity analysis of probabilistic inference w.r.t. a parameter by using the semiring of the polynomials with coefficients in $[0, 1]$. Further, AMC over GRAD can be employed in the context of parameter learning (Manhaeve et al. 2018), as it can produce the gradient w.r.t. a parameter. More applications and details can be found in (Kimmig, Van den Broeck, and De Raedt 2017).

Theorem 34. *AMC over $\mathcal{R}$ is Karp reducible to* SAT($\mathcal{R}$) *and vice versa, and thus* NP($\mathcal{R}$)*-complete w.r.t. Karp reductions for every semiring $\mathcal{R}$.*

Proof (sketch). $\Rightarrow$: We can translate T into a ΣBF β and weight it with α using $(v_i * \alpha(v_i) + \neg v_i * \alpha(\neg v_i))$ for $v_i \in V$. $\Leftarrow$: SAT($\mathcal{R}$) is a sum of products of the semiring values r that occur in the input ΣBF. We add for each occurrence r_i of r a variable v_r^i with $\alpha(v_r^i) = r, \alpha(\neg v_r^i) = e_{\otimes}$. Then we use T to specify which products are included in the sum. □

In combination with the relation of NP($\mathcal{R}$) to classical complexity classes these completeness results give us a better insight into the complexity of ΣFO-EVAL($\mathcal{R}$), SCSPs and AMC. Specifically, as SCSPs are only defined over c-semirings (which are idempotent), we see that over appropriately encoded, finitely generated semirings we stay in $\text{FP}_{\parallel}^{\text{NP}}$.

Conclusion & Outlook

On the one hand, the characterization of SUMPROD over $\mathcal{R}$ as NP($\mathcal{R}$)-complete shows that it is hard to solve, in most cases even significantly harder than SAT, as SRTMs work analogously to NTMs but retain more information via semiring values. Over $\mathbb{N}$ this is "only" the number of solutions, whereas over $\mathbb{N}[(x_i)_\infty]$ we can even obtain the number of solutions of all propositional formulas in n variables with one ΣBF of polynomial size. This explains why SUMPROD over $\mathbb{N}[(x_i)_\infty]$ is unlikely to be in FPSPACE(POLY).

On the other hand, the fact that we can use SRTMs to solve SUMPROD also provides upper-bounds on the hardness. We cannot compute the sum of arbitrarily generated semiring values but are restricted to sums of products of semiring values from a finite set or the input.

The investigation of the relation between the semiring classes NP($\mathcal{R}$) and classical complexity classes showed that NP($\mathbb{B}$), NP($\mathbb{N}$), NP($\mathcal{R}_{\max}$) correspond naturally to NP, #P, OPTP, respectively. We thus can see NP($\mathcal{R}$) as *a* possible generalization of NP to an algebraic setting. Alternative such generalizations would be of interest.

While SUMPROD is likely to be very hard (i.e., not in FPSPACE(POLY)) for some semirings, e.g. $\mathbb{N}[(x_i)_\infty]$, our results show that it can be solved using algorithms for #P (resp. NP) in a wide range of cases; viz. for many commutative, finitely generated (and resp. idempotent) semirings. We further demonstrated how our theorems facilitate the complexity analysis of specific semirings.

We characterized the complexity of multiple practically relevant problems: ΣFO-EVAL($\mathcal{R}$), computing the maximum achievable consistency of an SCSP over $\mathcal{R}$, and AMC over $\mathcal{R}$ are all NP($\mathcal{R}$)-complete. Together with our results that relate NP($\mathcal{R}$) to known complexity classes, this provides complexity results for a wide range of settings. Since the translations we used in the complexity proofs are quite natural, their benefit may even go beyond that. Although we have not considered this in detail, one may suspect that they allow to transfer the results for tractable fragments of AMC, SUMPROD etc., in (Bacchus, Dalmao, and Pitassi 2009; Kimmig, Van den Broeck, and De Raedt 2017; Friesen and Domingos 2016), from one framework to the other.

Outlook There are other formalisms, like semiring-induced propositional logic (Larrosa, Oliveras, and Rodríguez-Carbonell 2010) and Algebraic Derivation Counting (Green, Karvounarakis, and Tannen 2007) that are related to SUMPROD but do not seem to fall into this category of problems immediately. It would be interesting to see what the exact relationship is and what this means for their complexity. Extending our study to more semirings, e.g. to ones with SAT($\mathcal{R}$) in OPTP like the most probable explanation semiring $([0, 1], \max, \cdot, 0, 1)$, is another issue.

Also, SUMPROD problems can be naturally generalized to arbitrary stacks of sums and products using the Π quantifier of weighted QBFs. The extension of the definition of SRTMs seems feasible by using the approach of Ladner (1989) who introduced a counting version of the polynomial hierarchy based on alternating Turing Machines.

Acknowledgments

Thanks to the reviewers for their constructive comments. This work has been supported by FWF project W1255-N23 and by EU grant Humane AI Net ICT-48-2020-RIA / 952026 .

References

Aji, S. M.; and McEliece, R. J. 2000. The generalized distributive law. *IEEE transactions on Information Theory* 46(2): 325–343.

Bacchus, F.; Dalmao, S.; and Pitassi, T. 2009. Solving# SAT and Bayesian inference with backtracking search. *JAIR* 34: 391–442.

Bistarelli, S.; Montanari, U.; Rossi, F.; Schiex, T.; Verfaillie, G.; and Fargier, H. 1999. Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison. *Constraints* 4(3): 199–240.

Bistarelli, S.; and Santini, F. 2010. A common computational framework for semiring-based argumentation systems1, 2. In *ECAI*, volume 215, 131.

Cadoli, M.; Donini, F. M.; and Schaerf, M. 1996. Is intractability of nonmonotonic reasoning a real drawback? *AI* 88(1-2): 215–251.

Droste, M.; and Gastin, P. 2007. Weighted automata and weighted logics. *TCS* 380(1): 69.

Dudek, J. M.; Phan, V.; and Vardi, M. Y. 2020. ADDMC: Weighted Model Counting with Algebraic Decision Diagrams. In *AAAI*, 1468–1476.

Eisner, J. 2002. Parameter estimation for probabilistic finite-state transducers. In *ACL*, 1–8.

Eiter, T.; and Kiesel, R. 2020. ASP(AC): Answer Set Programming with Algebraic Constraints. *arXiv preprint arXiv:2008.04008* .

Fenner, S. A.; Fortnow, L. J.; and Kurtz, S. A. 1994. Gap-definable counting classes. *JCSS* 48(1): 116–148.

Friesen, A.; and Domingos, P. 2016. The sum-product theorem: A foundation for learning tractable models. In *ICML*, 1909–1918.

Ganian, R.; Kim, E. J.; Slivovsky, F.; and Szeider, S. 2018. Sum-of-Products with Default Values: Algorithms and Complexity Results. In Tsoukalas, L. H.; Grégoire, É.; and Alamaniotis, M., eds., *ICTAI*, 733–737. IEEE. doi:10.1109/ICTAI.2018.00115. URL http://www.ac.tuwien.ac.at/files/tr/ac-tr-18-007.pdf.

Gary, M. R.; and Johnson, D. S. 1979. Computers and Intractability: A Guide to the Theory of NP-completeness.

Goodman, J. 1999. Semiring parsing. *CL* 25(4): 573–605.

Green, T. J.; Karvounarakis, G.; and Tannen, V. 2007. Provenance semirings. In *ACM SIGMOD-SIGACT-SIGART*, 31–40. ACM.

Jenner, B.; and Torán, J. 1993. Computing functions with parallel queries to NP. In *Structure in Complexity Theory*, 280–291. IEEE.

Karp, R.; and Lipton, R. J. 1982. Turing machines that take advice. In *L'Enseign. Math.*, volume 28.

Kimmig, A.; Van den Broeck, G.; and De Raedt, L. 2011. An algebraic Prolog for reasoning about possible worlds. In *AAAI*.

Kimmig, A.; Van den Broeck, G.; and De Raedt, L. 2017. Algebraic model counting. *Journal of Applied Logic* 22: 46–62.

Krentel, M. W. 1988. The complexity of optimization problems. *JCSS* 36(3): 490–509.

Ladner, R. E. 1989. Polynomial space counting problems. *SICOMP* 18(6): 1087–1097.

Larrosa, J.; Oliveras, A.; and Rodríguez-Carbonell, E. 2010. Semiring-induced propositional logic: definition and basic algorithms. In *LPAR*, 332–347. Springer.

Mandrali, E.; and Rahonis, G. 2015. Weighted first-order logics over semirings. *Acta Cybernetica* 22(2): 435–483.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, 3749–3759.

Stearns, R. E.; and Hunt III, H. B. 1996. An algebraic model for combinatorial problems. *SICOMP* 25(2): 448–476.

Toda, S. 1989. On the computational power of PP and (+) P. In *FCS*, 514–519. IEEE Computer Society.

Tseitin, G. S. 1983. *On the Complexity of Derivation in Propositional Calculus*, 466–483. Berlin, Heidelberg: Springer Berlin Heidelberg.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SICOMP* 8(3): 410–421.

Van den Broeck, G.; Meert, W.; and Darwiche, A. 2014. Skolemization for weighted first-order model counting. In *KR*, 1–10.

Wilson, N. 2005. Decision diagrams for the computation of semiring valuations. In *IJCAI*, volume 5, 331–336.