Constraint Optimization over Semirings*

A. Pavan [1] ⓡ Kuldeep S. Meel, [2] ⓡ N. V. Vinodchandran [3] ⓡ Arnab Bhattacharyya[2]

[1] Iowa State University, USA, [2] National University of Singapore, Singapore, [3] University of Nebraska-Lincoln, USA

Abstract

Interpretations of logical formulas over semirings (other than the Boolean semiring) have applications in various areas of computer science including logic, AI, databases, and security. Such interpretations provide richer information beyond the truth or falsity of a statement. Examples of such semirings include Viterbi semiring, min-max or access control semiring, tropical semiring, and fuzzy semiring.

The present work investigates the complexity of constraint optimization problems over semirings. The generic optimization problem we study is the following: Given a propositional formula φ over n variable and a semiring $(K, +, \cdot, 0, 1)$, find the maximum value over all possible interpretations of φ over K. This can be seen as a generalization of the well-known satisfiability problem (a propositional formula is satisfiable if and only if the maximum value over all interpretations/assignments over the Boolean semiring is 1). A related problem is to find an interpretation that achieves the maximum value. In this work, we first focus on these optimization problems over the Viterbi semiring, which we call optConfVal and optConf.

We first show that for general propositional formulas in negation normal form, optConfVal and optConf are in $\mathrm{FP}^{\mathrm{NP}}$. We then investigate optConf when the input formula φ is represented in the conjunctive normal form. For CNF formulae, we first derive an upper bound on the value of optConf as a function of the number of maximum satisfiable clauses. In particular, we show that if r is the maximum number of satisfiable clauses in a CNF formula with m clauses, then its optConf value is at most $1/4^{m-r}$. Building on this we establish that optConf for CNF formulae is hard for the complexity class $\mathrm{FP}^{\mathrm{NP}[\log]}$. We also design polynomial-time approximation algorithms and establish an inapproximability for optConfVal. We establish similar complexity results for these optimization problems over other semirings including tropical, fuzzy, and access control semirings.

*The authors decided to forgo the old convention of alphabetical ordering of authors in favor of a randomized ordering, denoted by ⓡ. The publicly verifiable record of the randomization is available at https://www.aeaweb.org/journals/policies/random-author-order/search

1 Introduction

Classically, propositional formulae are interpreted over the Boolean semiring $\mathbb{B} = (\{\mathrm{F}, \mathrm{T}\}, \vee, \wedge, \mathrm{F}, \mathrm{T})$ which is the standard semantics for the logical truth. In this setting, the variables take one of the two values T (true) or F (false). However, it is natural to extend the semantics to other semirings. Here, the idea is to interpret logical formulae when the variables take values over a semiring $\mathbb{K} = (K, +, \cdot, 0, 1)$. Such interpretations provide richer information beyond the truth or falsity of a statement and have applications in several areas such as databases, AI, logic, and security (see (Imieliński and Lipski Jr 1989; Fuhr and Rölleke 1997; Zimányi 1997; Cui, Widom, and Wiener 2000; Cui 2002; Grädel and Tannen 2020) and references therein). In particular, semiring *provenance analysis* has been successfully applied in several software systems, such as Orchestra and Propolis (see, e.g., (Amsterdamer, Deutch, and Tannen 2011; Deutch et al. 2014; Foster, Green, and Tannen 2008; Green 2011; Tannen 2013)).

Examples of semirings that are studied in the literature include Viterbi semiring, fuzzy semiring, min-max or access control semiring, and tropical semiring. Semantics over the Viterbi semiring $\mathbb{V} = ([0, 1], \max, \cdot, 0, 1)$ has applications in database provenance, where $x \in [0, 1]$ is interpreted as a *confidence score* (Grädel and Tannen 2020; Green, Karvounarakis, and Tannen 2007; Tannen 2017; Grädel and Mrkonjic 2021), in probabilistic parsing, in probabilistic CSPs, and in Hidden Markov Models (Viterbi 1967; Klein and Manning 2003; Bistarelli, Montanari, and Rossi 1995). The access control semiring can be used as a tool in security specifications (Grädel and Tannen 2020). Other semirings of interest include the tropical semiring, used in cost analysis and algebraic formulation for shortest path algorithms (Mohri 2002), and fuzzy semirings used in the context of fuzzy CSPs (Bistarelli, Montanari, and Rossi 1995).

Optimization problems over Boolean interpretations have been central in many application as well as foundation areas. Indeed, the classical satisfiability problem is determining whether a formula $\phi(x_1, \cdots, x_n)$ has an interpretation/assignment over the Boolean semiring that evaluates to True. Even though semiring semantics naturally appear in a variety of applications, the optimization problems over semirings, other than the Boolean semiring, have not received much attention.

In this work, we introduce and investigate the complexity of optimization problems over semiring semantics. Let $\mathbb{K} = (K, +, \cdot, 0, 1)$ be a semiring with a total order over K and φ be a propositional formula over a set X of variables. A $\mathbb{K}$-interpretation π is a function from X to K. Such an interpretation can be naturally extended to formula φ, which we denote by $\mathsf{Sem}(\varphi, \pi)$. We study the following computational problem: Given a propositional formula φ in negation normal form over a set X of variables, compute the maximum value of $\mathsf{Sem}(\varphi, \pi)$ over all possible interpretations π. We call this problem optSemVal. A related problem, denoted optSem, is to compute an interpretation π that maximizes $\mathsf{Sem}(\varphi, \pi)$. Refer to Section 2 for a precise formulation of these problems.

There has been a rich history of work which formulated the notion of CSP over semirings and investigated local consistency algorithms in the general framework (Bistarelli 2004; Bistarelli and Gadducci 2006; Bistarelli, Montanari, and Rossi 1995, 1997; Bistarelli et al. 1999; Meseguer, Rossi, and Schiex 2006). These works did not involve interpretations and did not focus on the computational complexity of the above-defined problems. Relatedly, the computational complexity of sum-of-product problems over semirings has been studied recently (Eiter and Kiesel 2021). However, the problems they study are different from ours. To the best of our knowledge, optimization problems optSem and optSemVal that we consider over semirings have not been studied earlier and there are no characterizations of their computational complexity.

1.1 Our Results

We comprehensively study the computational complexity of optSem and the related problem optSemVal over various semirings such as Viterbi semiring, tropical semiring, access control semiring and fuzzy semiring, from both an algorithmic and a complexity-theoretic viewpoint. When the underlying semiring is the Viterbi semiring, we call these problems optConf and optConfVal. Our results can be summarized as follows:

1. We establish that both optConf and optConfVal are in the complexity class $\mathrm{FP}^{\mathrm{NP}}$. The crucial underlying observation is that even though π maps X to real values in the range $[0, 1]$; the solution to optConfVal can be represented using polynomially many bits. We then draw upon connections to Farey sequences to derive an algorithm with polynomially many NP calls (Theorem 3.2).
2. For CNF formulas, we establish an upper bound on optConfVal as a function of the number of maximum satisfiable clauses (Theorem 3.7).
3. We also establish a lower bound on the complexity of optConfVal and optConf. In particular, we show that both the problems are hard for the complexity class $\mathrm{FP}^{\mathrm{NP[log]}}$. To this end, we demonstrate a reduction from MaxSATVal to optConfVal; this reduction crucially relies on the above-mentioned upper bound on optConfVal in terms of the number of maximum satisfiable clauses (Theorem 3.9).
4. We design a polynomial-time approximation algorithm for optConfVal and establish an inapproximability result. In particular, for 3-CNF formulas with m clauses, we design a 0.716^m-approximation algorithm and show that the approximation factor can not be improved to 0.845^m unless P = NP (Theorems 4.3 and 4.4).
5. Finally, we show that for the access control semiring, the complexity of these optimization problems is equivalent to the corresponding problems over Boolean semiring (Theorem 5.3).

Remark 1. *Since Viterbi semiring and tropical semiring are isomorphic via the mapping $x \leftrightarrow -\ln x$, results established for Viterbi semiring also hold for the tropical semiring. Fuzzy semiring can be seen as an "infinite refinement" of access control semiring with the same algebraic structure, results that we establish for access control semiring also hold for fuzzy semiring.*

Organization. The rest of the paper is organized as follows. We give the necessary notation and definitions in Section 2. Section 3 details our results on the computational complexity of optConf and optConfVal. Section 4 deals with approximate algorithms and the hardness of approximation of optConfVal. In Section 5, we give complexity results for optimization problems for the access control semiring. Finally, we conclude in Section 6. *Due to space constraints, many of the involved proofs are omitted and will in the full version.*

2 Preliminaries

We assume that the reader is familiar with definition of a semiring. We denote a generic semiring by $\mathbb{K} = (K, +, \cdot, 0, 1)$ where K is the underlying set. For interpreting formulas over $\mathbb{K}$, we will add a "negation" function $\urcorner : K \to K$. We assume $\urcorner$ is a bijection so that $\urcorner(\urcorner(x)) = x$, and $\urcorner(0) = 1$. For ease of presentation, we use the most natural negation function (depending on the semiring). However, many of our results hold for very general interpretations of negation. Finally, as our focus is on optimization problems, we will also assume a (natural) total order on the elements of K.

For a set $X = \{x_1, x_2, \ldots x_n\}$ of variables, we associate the set $\overline{X} = \{\neg x_1, \ldots, \neg x_n\}$. We call $X \cup \overline{X}$ the literals and formulas we consider are propositional formulas over $X \cup \overline{X}$ in *negation normal form*. We also view a propositional formula φ in negation normal form as a rooted directed tree wherein each leaf node is labeled with a literal, 1, or 0 and each internal node is labeled with conjunction ($\wedge$) or disjunction $\vee$. Note that viewing φ as a tree ensures a similar size as its string representation. We call the tree representing the formula φ as *formula tree* and denote it with T_φ. For a propositional formula $\varphi(x_1, \cdots, x_n)$, in negation normal form we use m to denote the size of the formula, i.e. the total number of occurrences of each variable and its negation. When $\varphi(x_1, \cdots x_n)$ is in CNF form, m denotes the number of clauses.

We interpret a propositional formula over a semiring $\mathbb{K}$ by mapping the variables to K and naturally extending it. Formally, a $\mathbb{K}$-interpretation is a function $\pi : X \to K$. We extend π to an arbitrary propositional formula φ in negation normal form, which is denoted by $\mathsf{Sem}(\varphi, \pi)$ (Sem stands for 'semantics'), as follows.

- $\mathsf{Sem}(x, \pi) = \pi(x)$
- $\mathsf{Sem}(\neg x, \pi) = \urcorner(\pi(x))$
- $\mathsf{Sem}(\alpha \vee \beta, \pi) = \mathsf{Sem}(\alpha, \pi) + \mathsf{Sem}(\beta, \pi)$
- $\mathsf{Sem}(\alpha \wedge \beta, \pi) = \mathsf{Sem}(\alpha, \pi) \cdot \mathsf{Sem}(\beta, \pi)$

2.1 Optimization Problems and Complexity Classes

For a formula φ, we define $\mathsf{optSemVal}(\varphi)$ as

$$\mathsf{optSemVal}(\varphi) = \max_{\pi}\{\mathsf{Sem}(\varphi, \pi)\},$$

where $\max$ is taken over all possible $\mathbb{K}$-interpretations from X to K.

Definition 2.1 (optSem and optSemVal)**.** Given a propositional formula φ in negation normal form, the optSemVal problem is to compute $\mathsf{optSemVal}(\varphi)$. The optSem problem is to compute a $\mathbb{K}$-interpretation that achieves $\mathsf{optSemVal}(\varphi)$, i.e, output π^* so that $\mathsf{optSemVal}(\varphi) = \mathsf{Sem}(\varphi, \pi^*)$.

Notice that when $\mathbb{K}$ is the Boolean semiring (with $0 < 1$ ordering and standard negation interpretation), optSemVal is the well-known satisfiability problem: the formula φ is satisfiable if and only if $\mathsf{optSemVal}(\varphi) = 1$. Also, the problem optSem is to output a satisfying assignment if the formula φ is satisfiable.

In this work, we consider the following semirings.

1. Viterbi semiring $\mathbb{V} = ([0, 1], \max, \cdot, 0, 1)$. As mentioned, the Viterbi semiring has applications in database provenance, where $x \in [0, 1]$ is interpreted as confidence scores, in probabilistic parsing, in probabilistic CSPs, and in Hidden Markov Models.
2. The tropical semiring $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$. The tropical semiring is isomorphic to the Viterbi semiring via the mapping $x \leftrightarrow -\ln x$.
3. The fuzzy semiring $\mathbb{F} = ([0, 1], \max, \min, 0, 1)$.
4. Access control semiring $\mathbb{A}_k = ([k], \max, \min, 0, k)$. Intuitively, each $i \in [k]$ is associated with an access control level with natural ordering. Here 0 corresponds to public access and n corresponds to no access at all. $[k]$ is the set $\{0 < 1 < \cdots < k\}$.

Most of our focus will be on complexity of optSem and optSemVal problems over the Viterbi semiring. We call the corresponding computational problems optConf and optConfVal respectively. We call the extended interpretation function Sem as Conf in this case.

Definition 2.2 (MaxSat and MaxSatVal)**.** Given a propositional formula φ in CNF form, the MaxSat problem is to compute an assignment of φ that satisfies the maximum number of clauses. Given a propositional formula φ in CNF form, the MaxSatVal problem is to compute the maximum number of clauses of φ that can be satisfied.

We need a notion of reductions between functional problems. We use the notion of *metric reductions* introduced by Krentel (Krentel 1988).

Definition 2.3 (Metric Reduction)**.** For two functions $f, g : \{0, 1\}^* \to \{0, 1\}^*$, we say that f metric reduces to g if there are polynomial-time computable functions h_1 and h_2 where $h_1 : \{0, 1\}^* \to \{0, 1\}^*$ (the reduction function) and $h_2 : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ so that for any x, $f(x) = h_2(x, g(h_1(x)))$.

Definition 2.4. For a function $t : \mathbb{N} \to \mathbb{N}$, $\mathrm{FP}^{\mathrm{NP}[t(n)]}$ denotes the class of functions that can be solved in polynomial-time with $O(t(n))$ queries to an NP oracle where n is the size of the input. When $t(n)$ is some polynomial, we denote the class by $\mathrm{FP}^{\mathrm{NP}}$.

Metric reductions are used to define notions of completeness and hardness for function classes $\mathrm{FP}^{\mathrm{NP}}$ and $\mathrm{FP}^{\mathrm{NP}[\log]}$. The following result due to Krentel (Krentel 1988) characterizes the complexity of the MaxSatVal problem.

Theorem 2.5 ((Krentel 1988))**.** MaxSatVal *is complete for* $\mathrm{FP}^{\mathrm{NP}[\log]}$ *under metric reductions.*

The following proposition is a basic ingredient in our results. It can be proved using basic calculus.

Proposition 1. *Let $f(x) = x^a(1 - x)^b$ where a, b are non-negative integers, the maximum value of $f(x)$ over the domain $[0, 1]$ is attained when $x = \frac{a}{a+b}$. The maximum value of the function is $(\frac{a}{a+b})^a(\frac{b}{a+b})^b$.*

3 Computational Complexity of Confidence Maximization

For semantics over Viterbi semiring we assume the standard closed world semantics and use the negation function $\urcorner(x) = 1 - x$. Thus we have $\mathsf{Conf}(\neg x, \pi) + \mathsf{Conf}(x, \pi) = 1$. However, our upper bound proofs go through for any reasonable negation function. We discuss this in Remark 2.

Since $\mathsf{Conf}(\varphi, \pi)$ can be computed in polynomial time, optConf is at least as hard as optConfVal. The following observation states that computing optConfVal and optConf are NP-hard.

Observation 3.1. *For a formula φ,* $\mathsf{optConfVal}(\varphi) = 1$ *if and only if φ satisfiable. Hence both* optConf *and* optConfVal *are NP-hard.*

While both optConf and optConfVal are NP-hard, we would like to understand their relation to other maximization problems. In the study of optimization problems, the complexity classes $\mathrm{FP}^{\mathrm{NP}}$ and $\mathrm{FP}^{\mathrm{NP}[\log]}$ play a key role. In this section, we investigate both upper and lower bounds for these problems in relation to the classes $\mathrm{FP}^{\mathrm{NP}}$ and $\mathrm{FP}^{\mathrm{NP}[\log]}$.

3.1 An Upper Bound for General Formulae

We show that optConfVal and optConf can be computed in polynomial-time with oracle queries to an NP language.

Theorem 3.2. optConfVal *for formulas in negation normal form is in* $\mathrm{FP}^{\mathrm{NP}}$.

Proof Idea: In order to show that $\mathsf{optConfVal}$ is in $\mathrm{FP}^{\mathrm{NP}}$, we use a binary search strategy using a language in NP. One of the challenges is that the confidence value could potentially be any real number in $[0,1]$ and thus apriori we may not be able to bound the number of binary search queries. However, we first argue that for any formula φ on n variables and with size m, $\mathsf{optConf}(\varphi)$ is a fraction of the form A/B where $1 \leq A \leq B \leq 2^{nm \log m}$. Ordered fractions of such form are known as *Farey* sequence of order $2^{nm \log m}$ (denoted as $\mathcal{F}_{2^{nm \log m}}$). Thus our task is to do a binary search over $\mathcal{F}_{2^{nm \log m}}$ with time complexity $O(nm \log m)$. However, in general binary search for an unknown element in the Farey sequence $\mathcal{F}_N$ with time complexity $O(\log N)$ appears to be unknown. We overcome this difficulty by using an NP oracle to aid the binary search. We will give the details now.

Definition 3.3. Let $\varphi(x_1, \cdots, x_n)$ be a propositional formula in negation normal form with size m. Let T_φ be its formula tree. A proof tree T of T_φ is a subtree obtained by the following process: for every OR node v, choose one of the sub-trees of v. For every AND node v, keep all the subtrees.

Note that in a proof tree every OR node has only one child.

Definition 3.4. Let $\varphi(x_1, \cdots, x_n)$ be a propositional formula in negation normal form and let T be a proof tree. We define the *proof tree polynomial* p_T by inductively defining a polynomial for the subtree at every node v (denoted by p_v): If the node v is a variable x_i, the polynoimal is x_i and if it is $\neg x_i$, the polynomial is $(1 - x_i)$. If v is an AND node with children $v_1, \ldots, v_s$, then $p_v = \prod_{i=1}^{s} p_s$. If v is an OR node with a child u, then $p_v = p_u$.

Claim 3.4.1. *Let $\varphi(x_1, \cdots, x_n)$ be a propositional formula in negation normal form and let T be a proof tree of φ.*

1. *The proof tree polynomial p_T is of the form*

$$\prod_{i=1}^{n} x_i^{a_i}(1 - x_i)^{b_i}$$

where $0 \leq a_i + b_i \leq m$.

2. *For a $\mathbb{V}$-interpretation π,*

$$\mathsf{Conf}(T, \pi) = p_T\left(\pi(x_1), \ldots, \pi(x_n)\right).$$

3. *Both $\mathsf{optConf}(T)$ and $\mathsf{optConfVal}(T)$ can be computed in polynomial-time.*
4. $\mathsf{optConfVal}(T) = \Pi_{i=1}^{n}\left(\frac{a_i}{a_i+b_i}\right)^{a_i}\left(\frac{b_i}{a_i+b_i}\right)^{b_i}$.

The next claim relates $\mathsf{optConf}$ of the formula φ to $\mathsf{optConf}$ of its proof trees. The proof of this claim follows from the definition of proof tree and standard induction.

Claim 3.4.2. *For a formula φ,*

$$\mathsf{optConfVal}(\varphi) = \max_T \mathsf{optConfVal}(T)$$

where maximum is taken over all proof trees T of T_φ. If T^* is the proof tree for which $\mathsf{optConf}(T)$ is maximized, then $\mathsf{optConf}(T^*) = \mathsf{optConf}(\varphi)$.

The above claim states that $\mathsf{optConf}(\varphi)$ can be computed by cycling through all proof trees T of φ and computing $\mathsf{optConf}(T)$. Since there could be exponentially many proof trees, this process would take exponential time. Our task is to show that this process can be done in $\mathrm{FP}^{\mathrm{NP}}$. To do this we establish a claim that restricts values that $\mathsf{optConfVal}(\varphi)$ can take. We need the notion of *Farey sequence*.

Definition 3.5. For any positive integer N, the *Farey sequence* of order N, denoted by $\mathcal{F}_N$, is the set of all irreducible fractions p/q with $0 < p < q \leq N$ arranged in increasing order.

Claim 3.5.1. *1. For a propositional formula $\varphi(x_1, \cdots, x_n)$, $\mathsf{optConfVal}(\varphi)$ belongs to the Farey sequence $\mathcal{F}_{2^{nm \log m}}$.*

2. For any two fractions u and v from $\mathcal{F}_{2^{nm \log m}}$, $|u - v| \geq 1/2^{2nm \log m}$

Consider the following language

$$L_{opt} = \{\langle \varphi, v\rangle \mid \mathsf{optConfVal}(\varphi) \geq v\}$$

Claim 3.5.2. *L_{opt} is in* NP.

We need a method that given two fractions u and v and an integer N, outputs a fraction $p/q : u \leq p/q \leq v$, and $p/q \in \mathcal{F}_N$. We give an $\mathrm{FP}^{\mathrm{NP}}$ algorithm that makes $O(N)$ queries to the NP oracle to achieve this. We first define the NP language L_{farey}. For this we fix any standard encoding of fraction using the binary alphabet. Such an encoding will have $O(\log N)$ bit representation for any fraction in $\mathcal{F}_N$.

$$L_{farey} = \{\langle N, u, v, z\rangle \mid \exists z'; u \leq zz' \leq v \;\&\; zz' \in \mathcal{F}_N\}$$

The following claim is easy to see.

Claim 3.5.3. $L_{farey} \in$ NP.

Now we are ready to prove the Theorem 3.2.

Proof. (*of Theorem 3.2*). The algorithm performs a binary search over the range $[0,1]$ by making adaptive queries $\langle \varphi, v\rangle$ to the NP language L_{opt} starting with $v = 1$. At any iteration of the binary search, we have an interval $I = [I_l, I_r]$ and with the invariant $I_l \leq \mathsf{optConfVal}(\varphi) < I_r$. The binary search stops when the size of the interval $[I_l, I_r] = 1/2^{2nm \log m}$. Since each iteration of the binary search reduces the size of the interval by a factor of 2, the search stops after making $2nm \log m$ queries to L_{opt}. The invariant ensures that $\mathsf{optConfVal}(\varphi)$ is in this interval. Moreover, $\mathsf{optConfVal}(\varphi) \in \mathcal{F}_{2^{nm \log m}}$ (by item (1) of Claim 3.5.1) and there are no other fractions from $\mathcal{F}_{2^{nm \log m}}$ in this interval (by item (2) of Claim 3.5.1). Now, by making $O(nm \log m)$ queries to L_{farey} with $N = 2^{nm \log m}$, $u = I_l$, $v = I_r$, we can construct the binary representation of the unique fraction in $\mathcal{F}_{2^{nm \log m}}$ that lies between I_l and I_r which is $\mathsf{optConfVal}(\varphi)$. ☐

Next we show the optimal $\mathbb{V}$-interpretation can also be computed in polynomial time with queries to an NP oracle.

Theorem 3.6. $\mathsf{optConf}$ *for formulas in negation normal form can be computed in* $\mathrm{FP}^{\mathrm{NP}}$.

Proof. Let φ be a propositional formula in negation normal form. We use a prefix search over the encoding of proof trees of φ using an NP language to isolate a proof tree T such that $\mathsf{optConfVal}(\varphi) = \mathsf{optConfVal}(T)$. For this, we fix an encoding of proof trees of φ. Consider the following NP language L_{pt}:

$$\{\langle \varphi, v, z\rangle \mid \exists z' : zz' \text{encodes a proof tree } T \text{ of } \varphi \\ \& \ \mathsf{optConfVal}(T) = v\}$$

Claim 3.6.1. *L_{pt} is in* NP.

To complete the proof Theorem 3.6, given a propositional formula φ, we first use $\mathrm{FP}^{\mathrm{NP}}$ algorithm from Theorem 3.2 to compute $v^* = \mathsf{optConfVal}(\varphi)$. Now we can construct a proof tree T of φ so that $\mathsf{optConfVal}(T) = v^*$ by a prefix search using language L_{pt}. Now by Claim 3.4.1, we can compute a $\mathbb{V}$-interpretation π^* so that $\mathsf{Conf}(T, \pi^*) = v^*$. Thus π^* is an optimal $\mathbb{V}$-interpretation for φ, by Claim 3.4.2. □

Remark 2. *We revisit the semantics of negation. As stated earlier, by assuming the closed world semantics, we have $\urcorner(x) = 1 - x$. We note that this assumption is not strictly necessary for the above proof to go through. Recall that Item (1) of Claim 3.4.1 states that the proof tree polynomial is of the form $\prod x_i^{a_i}(1 - x_i)^{b_i}$. For a general negation function $\urcorner$, the proof tree polynomial is of the form $\prod x_i^{a_i}(\urcorner(x_i))^{b_i}$. Now if the maximum value of a term $x^a(\urcorner(x))^b$ can be found, for example when $\urcorner$ is an explicit differentiable function, the result will hold.*

3.2 Relation to MaxSat for CNF Formulae

In this section we study the $\mathsf{optConfVal}$ problem for CNF formulae and establish its relation to the MaxSat problem. We first exhibit an upperbound on the $\mathsf{optConfVal}(\varphi)$ using the maximum number of satisfiable clauses. Building on this result, in Section 3.3 we show that $\mathsf{optConfVal}$ for CNF formulae is hard for the complexity class $\mathrm{FP}^{\mathrm{NP}[\log]}$.

We first define some notation that will be used in this and next subsections. Let $\varphi(x_1, \cdots x_n) = C_1 \wedge \cdots \wedge C_m$ be a CNF formula and let π^* be an optimal $\mathbb{V}$-interpretation. For each clause C from φ, let $\pi^*(C)$ be the value achieved by this interpretation, i.e $\pi^*(C) = \mathsf{Conf}(C, \pi^*)$. Observe that since C is a disjunction of literals, $\pi^*(C) = \max_{\ell \in C}\{\pi^*(\ell)\}$. For a clause C, let

$$\ell_C = \operatorname{argmax}_{\ell \in C}\{\pi^*(\ell)\}$$

In the above, if there are multiple maximums, we take the smallest literal as ℓ_C (By assuming an order $x_1 < \neg x_1 < x_2 < \neg x_2 \cdots < x_n < \neg x_n$). Observe that, since we are working over the Viterbi semiring, $\mathsf{Conf}(C, \pi^*) = \pi^*(\ell_C)$. A literal ℓ is *maximizing literal for a clause C*, if $\ell_C = \ell$.

Since φ is a CNF formula, for any $\mathbb{V}$-interpretation π $\mathsf{Conf}(\varphi, \pi)$ is of the form $\Pi_{i=1}^m \mathsf{Conf}(C_i, \pi)$. Given a collection of clauses $\mathcal{D}$ from φ, the *contribution of $\mathcal{D}$ to* $\mathsf{Conf}(\varphi, \pi)$ is defined as $\Pi_{c \in \mathcal{D}} \mathsf{Conf}(C, \pi)$.

The following theorem provides an upperbound on $\mathsf{optConfVal}(\varphi)$ using $\mathsf{MaxSatVal}$. This is the main result of this section.

Theorem 3.7. *Let $\varphi(x_1, \cdots, x_n)$ be a CNF formula with m clauses. Let r be the maximum number of clauses that can be satisfied. Then* $\mathsf{optConfVal}(\varphi) \leq 1/4^{(m-r)}$.

Proof. Let π^* be an optimal $\mathbb{V}$-interpretation for φ. A clause C is called *low-clause* if $\pi^*(C) < 1/2$, C is called a *high-clause* of $\pi^*(C) > 1/2$, and C is a *neutral-clause* if $\pi^*(C) = 1/2$. Let L, H, and N respectively denote the number of low, high, and neutral clauses.

We start with the following claim that relates the number of neutral clauses and the number of high-clauses to r.

Claim 3.7.1. $\frac{N}{2} + H \leq r$

Proof. Suppose that the number of low-clauses is strictly less than $m - r$, thus number of high-clauses is more than r.

For a variable x, let

$$p_x = |\{C \mid C \text{ is neutral and } \ell_C = x\}|$$

and

$$q_x = |\{C \mid C \text{ is neutral and } \ell_C = \neg x\}|$$

That is p_x is the number of neutral clauses for which x is the maximizing literal and q_x is the number of neutral clauses for which $\neg x$ is the maximizing literal.

Consider the truth assignment that is constructed based on the following three rules: (1) For every high-clause C, set ℓ_C to True and $\neg \ell_C$ to False, 2) For every variable x, if one of p_x or q_x is not zero, then if $p_x \geq q_x$, then set x to True, otherwise set x to False. (3) All remaining variables are consistently assigned arbitrary to True/False values.

We argue that this is a consistent assignment: I.e, for every literal ℓ, both ℓ and $\neg \ell$ are not assigned the same truth value. Consider a literal ℓ. If there is a high clause C such that $\ell = \ell_C$, then this literal is assigned truth value True and $\neg \ell$ is assigned False. In this case, since $\pi^*(\ell) > 1/2$, $\pi^*(\neg \ell) < 1/2$. Thus $\neg \ell$ can not be maximizing literal for any high-clause and thus Rule (1) does not assign True to $\neg \ell$. Again, since $\pi^*(\ell) > 1/2$, there is no neutral-clause D such that $\ell = \ell_D$ or $\neg \ell = \ell_D$. Thus Rule (2) does not assign a truth value to either of ℓ or $\neg \ell$. Since ℓ and $\neg \ell$ are assigned truth values, Rule (3) does not assign a truth value to ℓ or $\neg \ell$.

Consider a variable x where at least one of p_x or q_x is not zero. In this case x or $\neg x$ is maximizing literal for a neutral clause. Thus $\pi^*(x) = \pi^*(\neg x) = 1/2$ and neither x nor $\neg x$ is maximizing literal for a high-clause. Thus Rule (1) does not assign a truth value to x or $\neg x$. Now x is True if and only if $p_x \geq q_x$, thus the truth value assigned to x (and $\neg x$) is consistent. Since Rule (3) consistently assigns truth values of literals that are not covered by Rules (1) and (2), the constructed assignment is a consistent assignment.

For every high clause C, literal ℓ_C is set to true. Thus the assignment satisfies all the high-clauses. Consider a variable x and let $\mathcal{D}$ be the (non-empty) collection of neutral clauses for which either x or $\neg x$ is a maximizing literal. As x is assigned True if and only if $p_x \geq q_x$, at least half the clauses from $\mathcal{D}$ are satisfied. Thus this assignment satisfies at least $H + \frac{N}{2}$ clauses. Since r is the maximum number of satisfiable clauses, the claim follows. □

For a literal ℓ, let a_ℓ be the number of low-clauses C for which ℓ is a maximizing literal, i.e,

$$a_\ell = |\{C \mid C \text{ is a low-clause and } \ell_C = \ell\}|,$$

and

$$b_\ell = |\{C \mid C \text{ is a high-clause and } \ell_C = \neg\ell\}|,$$

We show the following relation between a_ℓ and b_ℓ.

Claim 3.7.2. *For every literal ℓ, $a_\ell \leq b_\ell$.*

We next bound the contribution of neutral and low clauses to $\mathsf{optConfVal}(\varphi)$. For every neutral clause C, $\pi^*(C) = 1/2$, thus we have the following observation.

Observation 3.8. *The contribution of neutral clauses to* $\mathsf{Conf}(\varphi, \pi^*)$ *is exactly* $1/2^N$.

We establish the following claim.

Claim 3.8.1.

$$\mathsf{Conf}(\varphi, \pi^*) = \prod_\ell \left(\pi^*(\ell)^{a_\ell} \times (1-\pi^*(\ell))^{b_\ell}\right) \times \frac{1}{2^N}$$

Finally, we are ready to complete the proof of Theorem 3.7. For every literal ℓ, By Claim 3.7.2, $a_\ell \leq b_\ell$. Let $b_\ell = a_\ell + c_\ell$, $c_\ell \geq 0$. Consider the following inequalities.

$$\begin{aligned}
\mathsf{optConfVal}(\varphi) &= \mathsf{Conf}(\varphi, \pi^*) \\
&= \prod_\ell \left(\pi^*(\ell)^{a_\ell} \times (1-\pi^*(\ell))^{b_\ell}\right) \times \frac{1}{2^N} \\
&= \prod_\ell \left(\pi^*(\ell)^{a_\ell} \times (1-\pi^*(\ell))^{a_\ell + c_\ell}\right) \times \frac{1}{2^N} \\
&\leq \prod_\ell \left(\pi^*(\ell)^{a_\ell} \times (1-\pi^*(\ell))^{a_\ell}\right) \times \frac{1}{2^N} \\
&\leq \prod_\ell \left(\frac{1}{4^{a_\ell}}\right) \times \frac{1}{2^N} = \frac{1}{4^{L+N/2}}
\end{aligned}$$

In the above, equality at line 2 is due to Claim 3.8.1. The inequality at line 4 follows because $(1-\pi^*(\ell)) \leq 1$. The last inequality follows because $x(1-x)$ is maximized at $x = 1/2$. The last equality follows as $\sum a_\ell = L$. Note that the number of clauses $m = N + H + L$ and by Claim 3.7.1 $H + N/2 \leq r$. It follows that $L + N/2 \geq m - r$. Thus $\mathsf{optConfVal}(\varphi) = \mathsf{Conf}(\varphi, \pi^*) \leq \frac{1}{4^{L+N/2}} \leq \frac{1}{4^{m-r}}$. □

3.3 $\mathrm{FP}^{\mathrm{NP[log]}}$- Hardness

In this subsection, we show that $\mathsf{optConfVal}$ is hard for the class $\mathrm{FP}^{\mathrm{NP[log]}}$. We show this by reducing $\mathsf{MaxSatVal}$ to $\mathsf{optConfVal}$. Since $\mathsf{MaxSatVal}$ is complete for $\mathrm{FP}^{\mathrm{NP[log]}}$, the result follows. We also show that the same reduction can be used to compute a MaxSat assignment from an optimal $\mathbb{V}$-interpretation.

Theorem 3.9. $\mathsf{MaxSatVal}$ *metric reduces to* $\mathsf{optConfVal}$ *for CNF formulae. Hence* $\mathsf{optConfVal}$ *is hard for* $\mathrm{FP}^{\mathrm{NP[log]}}$ *for CNF formulae.*

Proof. Let $\varphi(x_i, \ldots, x_n) = C_1 \wedge \ldots \wedge C_m$ be a formula with m clauses on variables $x_1, \ldots, x_n$. Consider the formula φ' with m additional variables $y_1, \ldots, y_m$ constructed as follows: For each clause C_i of φ, add the clause $C_i' = C_i \vee y_i$ in φ'. Also add m unit clauses $\neg y_i$. That is

$$\varphi' = (C_1 \vee y_1) \wedge \ldots \wedge (C_m \vee y_m) \wedge \neg y_1 \wedge \cdots \wedge \neg y_m$$

Claim 3.9.1. $\mathsf{optConfVal}(\varphi') = \frac{1}{4^{m-r}}$ *where r is the maximum number of clauses that can be satisfied in φ.*

Proof. We show this claim by first showing that $\mathsf{optConfVal}(\varphi') \leq \frac{1}{4^{m-r}}$ and exhibiting an interpretation π^* so that $\mathsf{Conf}(\varphi, \pi^*) = \frac{1}{4^{m-r}}$. We claim that if r is the maximum number of clauses that can be satisfied in φ, then $m + r$ is the maximum number of clauses that can be satisfied in φ'. We will argue this by contradiction. Let $\mathbf{a}$ be an assignment that satisfies $> m + r$ clause in φ'. Let s be the number of y_is that are set to False. This assignment will satisfy $m - s$ clauses of the form $C_i \vee y_i$. However the total number of clauses of the form $C_i \vee y_i$ that are satisfied is $> m + r - s$. Thus there are $> r$ clauses of the form $C_i \vee y_i$ that are satisfied where y_i is set to False. This assignment when restricted to x_is will satisfy more than r clauses of φ. Hence the contradiction.

Thus from Theorem 3.7, it follows that $\mathsf{optConfVal}(\varphi') \leq \frac{1}{4^{m-r}}$. Now we exhibit an interpretation π^* so that $\mathsf{Conf}(\varphi, \pi^*) = \frac{1}{4^{m-r}}$. Consider an assignment $\mathbf{a} = a_1, \ldots, a_n$ for φ that satisfies r clauses. Consider the following interpretation π^* over the variable of φ': $\pi^*(x_i) = 1$ if $a_i =$ True and $\pi^*(x_i) = 0$ if $a_i =$ False. $\pi^*(y_i) = 0$ if and only if C_i is satisfied by $\mathbf{a}$. Else $\pi^*(y_i) = 1/2$. For every satisfiable clause C_i, $\mathsf{Conf}(C_i \vee y_i, \pi^*) = 1$ and $\mathsf{Conf}(\neg y_i, \pi^*) = 1$. For all other clauses C in φ', $\mathsf{Conf}(C, \pi^*) = 1/2$. Since there are r clauses that are satisfied, the number of clauses for which $\mathsf{Conf}(C, \pi^*) = 1/2$ is $2m - 2r$. Hence the $\mathsf{Conf}(\varphi', \pi^*) = \frac{1}{4^{(m-r)}}$. Thus $\mathsf{optConfVal}(\varphi') = \frac{1}{4^{m-r}}$. □

Since $\mathsf{optConfVal}(\varphi') = 1/4^{m-r}$, $\mathsf{MaxSatVal}$ for φ can be computed by knowing the $\mathsf{optConfVal}$. □

While the above theorem shows that $\mathsf{MaxSatVal}$ can be computed from $\mathsf{optConfVal}$, the next theorem shows that a maxsat assignment can be computed from an optimal $\mathbb{V}$-interpretation.

Theorem 3.10. MaxSat *metric reduces to* $\mathsf{optConf}$.

Proof. Consider the same reduction as from the previous theorem. Our task is to construct a MaxSat assignment for φ, given an optimal $\mathbb{V}$-interpretation π for φ'. By the earlier theorem, $\mathsf{Conf}(\varphi', \pi) = \frac{1}{4^{m-r}}$, where r is the maximum number of satisfiable clauses of φ. We first state a set of claims without proof.

Claim 3.10.1. *For every i, if y_i is not maximizing literal for clause C_i', then $\pi(y_i) = 0$.*

Claim 3.10.2. *For all y_i; $\pi(y_i) \in \{0, 1/2\}$.*

Claim 3.10.3. *For all x_i, if x_i or $\neg x_i$ is a maximizing literal, then $\pi(x_i) \in \{0, 1, 1/2\}$*

Claim 3.10.4. *For every x_i with $\pi(x_i) = 1/2$, x_i and $\neg x_i$ are maximizing literals for exactly the same number of clauses.*

We will show how to construct a MaxSat assignment from π: If $\pi(x_i) = 0$, set the truth value of x_i to False, else set the truth value of x_i to True.

By Claim 3.10.3, $\pi(x_i) = \{0, 1/2, 1\}$. Let H be the number of clauses for which maximizing literal ℓ is a x-variable and $\pi(\ell) = 1$. Note that the above truth assignment will satisfy all the H clauses. Let N be number of clauses for which maximizing literal ℓ is a x-variable and $\pi(\ell) = 1/2$. By Claim 3.10.4, in exactly $N/2$ clauses a positive literal is maximizing, and thus all these $N/2$ clauses are satisfied by our truth assignment. Thus the total number of clauses satisfied by the truth assignment is $N/2+H$. Let Y the number of clauses in which y_i is maximizing literal. By Claim 3.10.2, $\pi(y_i) = 1/2$ when y_i is maximizing literal. Thus

$$\mathsf{Conf}(\varphi', \pi) = 1^H \times (\frac{1}{2})^N \times (\frac{1}{2})^{2Y} = \frac{1}{4^{N/2+Y}} = \frac{1}{4^{m-r}}$$

The last equality follows from Claim 3.9.1. Thus $m - r = N/2 + Y$, combining this with $m = H + N + Y$, we obtain that $N/2 + H = r$. Thus the truth assignment constructed will satisfy r clauses and is thus a MaxSat assignment. □

4 Approximating optConfVal

We study the problem of approximating optConfVal efficiently. Below, a k-SAT formula is a CNF formula with *exactly* k distinct variables in any clause. We start with the following proposition.

Lemma 4.1. *Let $a_1, \cdots a_n$ be an assignment, that satisfies r clauses of a CNF formula $\varphi(x_1, \cdots x_n)$. There is an interpretation π so that* $\mathsf{Conf}(\varphi, \pi)$ *is* $\left(\frac{m-r}{m}\right)^{m-r} \left(\frac{r}{m}\right)^r$

Hence, for example, if φ is a 3-SAT formula, since a random assignment satisfies $7/8$ fraction of the clauses in expectation, for a random assignment $r \geq 7m/8$, and by Lemma 4.1, $\mathsf{optConfVal}(\varphi) > 0.686^m$. The following lemma shows that one can get a better lower bound on optConfVal in terms of the clause sizes for CNF formulae.

Lemma 4.2. *For every CNF formula φ,* $\mathsf{optConfVal}(\varphi) \geq e^{-\sum_i \frac{1}{k_i}}$ *where k_i is the arity of the i'th clause in φ.*

This yields a probabilistic algorithm. For example, if φ is a 3-SAT formula, $\mathsf{optConfVal}(\varphi) > 0.716^m$ and thus improving on the result of Lemma 4.1. In fact, we can design a deterministic polynomial time algorithm that finds an interpretation achieving the trust value guaranteed by Lemma 4.2, using the well-known 'method of conditional expectation' to derandomize the construction in the proof (For example, see (Alon and Spencer 2008; Goemans and Williamson 1994)).

Theorem 4.3. *There is a polynomial-time, $e^{-m/k}$-approximation algorithm for* optConf*, when the input formulas are k-CNF formulas with m-clauses.*

Next, we show that the approximation factor $e^{-m/k}$ can not be significantly improved.

Theorem 4.4. *There is no polynomial-time $\frac{1}{4^{m(2^{-k}-\varepsilon)}}$-approximation algorithm for* optConf *for k-SAT formulae, unless* $\mathrm{P} = \mathrm{NP}$.

Thus, for example for 3-SAT formulas, while we have a polynomial-time, 0.716^m-approximation algorithm (by Theorem 4.3), we cannot expect an efficient 0.845^m-approximation algorithm by the above result unless P equals NP. It remains an interesting open problem to determine the optimal approximation ratio for this problem achievable by a polynomial time algorithm.

5 Complexity of Access Maximization

In this section, we study the optimization problems for the access control semiring $\mathbb{A}_k = ([k], \max, \min, 0, k)$. We refer to the corresponding computational problems as optAccessVal and optAccess. For this section we first assume the negation function is the additive inverse modulo k. That is $\urcorner(a) = b$ such that $a + b \equiv 0 \pmod{k}$.

Theorem 5.1. *Let $\varphi(x_1, \cdots x_n)$ be a propositional formula in negation normal form and $\mathbb{A}_k = ([k], \max, \min, 0, k)$. The following statement holds.*

- *If φ is satisfiable, then* $\mathsf{optAccessVal}(\varphi) = k$.
- *If φ is not satisfiable, then* $\mathsf{optAccessVal}(\varphi) = \lfloor \frac{k}{2} \rfloor$.

For a general negation function, we can establish an analogous theorem. For this, we define the notion of the *index of negation*. Given a negation function $\urcorner$, its index denoted by $Index(\urcorner)$ is the largest ℓ for which there exists $a \in [k]$, such that both a and $\urcorner(a)$ are at least ℓ.

Theorem 5.2. *Let $\varphi(x_1, \cdots x_n)$ be a propositional formula in negation normal form and $\mathbb{A}_k = ([k], \max, \min, 0, k)$. The following statement holds.*

- *If φ is satisfiable, then* $\mathsf{optAccessVal}(\varphi) = k$.
- *If φ is not satisfiable, then* $\mathsf{optAccessVal}(\varphi) = Index(\urcorner)$.

The following is a corollary to the above result and its proof which states that the complexity of optimization problems over access control semiring is equivalent to their complexity over the Boolean semiring.

Theorem 5.3. *The problem* optAccessVal *and* SAT *are equivalent under metric reductions. Similarly, the problem* optAccess *and the problem of computing a satisfying assignment of a given Boolean formula are equivalent under metric reductions.*

6 Conclusion

In this work, we provided a comprehensive study of the computational complexity of optSem and the related problem optSemVal over various semirings such as Viterbi semiring, tropical semiring, access control semiring and fuzzy semiring, from both an algorithmic and a complexity-theoretic viewpoint. An exciting recent development in the field of CSP/SAT solving has been the development of solvers for LexSAT, which seeks to find the smallest lexicographic satisfying assignment of a formula (Marques-Silva et al. 2011). In this regard, Theorem 3.2 opens up exciting directions of future work to develop efficient techniques for optConf.

7 Acknowledgements

We thank the anonymous reviewers of AAAI-23 for valuable comments. This research is supported by the National Research Foundation under the NRF Fellowship Programme[NRF-NRFFAI1-2019-0004] and Campus for Research Excellence and Technological Enterprise (CREATE) programme. Bhattacharyya was supported in part by the NRF Fellowship Programme [NRF-NRFFAI1-2019-0002] and an Amazon Research Award. Vinod was supported in part by NSF CCF-2130608 and NSF HDR:TRIPODS-1934884 awards. Pavan was supported in part by NSF CCF-2130536, and NSF HDR:TRIPODS-1934884 awards.

References

Alon, N.; and Spencer, J. H. 2008. *The Probabilistic Method, Third Edition*. Wiley-Interscience series in discrete mathematics and optimization. Wiley.

Amsterdamer, Y.; Deutch, D.; and Tannen, V. 2011. Provenance for aggregate queries. In *Proc. of PODS*, 153–164.

Bistarelli, S. 2004. *Semirings for soft constraint solving and programming*, volume 2962. Springer Science & Business Media.

Bistarelli, S.; and Gadducci, F. 2006. Enhancing constraints manipulation in semiring-based formalisms. In *ECAI*, volume 141, 63–67.

Bistarelli, S.; Montanari, U.; and Rossi, F. 1995. Constraint solving over semirings. In *IJCAI (1)*, 624–630. Citeseer.

Bistarelli, S.; Montanari, U.; and Rossi, F. 1997. Semiring-based constraint satisfaction and optimization. *J. ACM*, 44(2): 201–236.

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.

Cui, Y. 2002. *Lineage tracing in data warehouses*. Ph.D. thesis, Stanford University.

Cui, Y.; Widom, J.; and Wiener, J. L. 2000. Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems (TODS)*, 25(2): 179–227.

Deutch, D.; Milo, T.; Roy, S.; and Tannen, V. 2014. Circuits for Datalog Provenance. In *Proc. of ICDT*, 201–212. OpenProceedings.org.

Eiter, T.; and Kiesel, R. 2021. On the Complexity of Sum-of-Products Problems over Semirings. In *Proc. of AAAI*, 6304–6311. AAAI Press.

Foster, J. N.; Green, T. J.; and Tannen, V. 2008. Annotated XML: queries and provenance. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 271–280.

Fuhr, N.; and Rölleke, T. 1997. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems (TOIS)*, 15(1): 32–66.

Goemans, M. X.; and Williamson, D. P. 1994. New 3/4-Approximation Algorithms for the Maximum Satisfiability Problem. *SIAM J. Discret. Math.*, 7(4): 656–666.

Grädel, E.; and Mrkonjic, L. 2021. Elementary Equivalence Versus Isomorphism in Semiring Semantics. In *Proc. of ICALP*, volume 198 of *LIPIcs*, 133:1–133:20.

Grädel, E.; and Tannen, V. 2020. Provenance analysis for logic and games. *Moscow Journal of Combinatorics and Number Theory*, 9(3): 203 – 228.

Green, T. J. 2011. Containment of conjunctive queries on annotated relations. *Theory of Computing Systems*, 49(2): 429–459.

Green, T. J.; Karvounarakis, G.; and Tannen, V. 2007. Provenance semirings. In *Proc. of PODS*, 31–40.

Imieliński, T.; and Lipski Jr, W. 1989. Incomplete information in relational databases. In *Readings in Artificial Intelligence and Databases*, 342–360. Elsevier.

Klein, D.; and Manning, C. D. 2003. A* Parsing: Fast Exact Viterbi Parse Selection. In Hearst, M. A.; and Ostendorf, M., eds., *Proc. of HLT-NAACL*. The Association for Computational Linguistics.

Krentel, M. W. 1988. The Complexity of Optimization Problems. *J. Comput. Syst. Sci.*, 36(3): 490–509.

Marques-Silva, J.; Argelich, J.; Graça, A.; and Lynce, I. 2011. Boolean lexicographic optimization: algorithms & applications. *Annals of Mathematics and Artificial Intelligence*, 62(3): 317–343.

Meseguer, P.; Rossi, F.; and Schiex, T. 2006. Soft constraints. In *Foundations of Artificial Intelligence*, volume 2, 281–328. Elsevier.

Mohri, M. 2002. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *J. Autom. Lang. Comb.*, 7(3): 321–350.

Tannen, V. 2013. Provenance propagation in complex queries. In *In Search of Elegance in the Theory and Practice of Computation*, 483–493. Springer.

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

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2): 260–269.

Zimányi, E. 1997. Query evaluation in probabilistic relational databases. *Theoretical Computer Science*, 171(1-2): 179–219.