Circuit Bounds for Conjunctive Queries with Self-joins

AUSTEN Z. FAN, University of Wisconsin–Madison, USA
PARASCHOS KOUTRIS, University of Wisconsin–Madison, USA
HANGDONG ZHAO, University of Wisconsin–Madison, USA

In this paper, we study circuit size bounds for Conjunctive Queries (CQs) under different semiring semantics. Recent work established tight bounds for self-join-free CQs over the tropical semiring, among other results [16]. Here, we extend these results in two main directions. First, we prove a lower bound for any self-join-free CQ over the Boolean semiring by extending Razborov and Alon-Boppana's classic lower bound result for the k-clique problem [1, 38]. Second, we characterize the circuit complexity of CQs with self-joins by relating them to appropriate self-join free CQs. Interestingly, such correspondence crucially depends on the underlying semiring. To achieve this result, we present a novel technique of investigating the circuit complexity of a CQ with self-joins through the lens of endomorphisms.

CCS Concepts: • **Theory of computation → Database theory**.

Additional Key Words and Phrases: Conjunctive Queries, Sum-product Polynomials, Circuit Size Bounds, Semirings

ACM Reference Format:
Austen Z. Fan, Paraschos Koutris, and Hangdong Zhao. 2025. Circuit Bounds for Conjunctive Queries with Self-joins. *Proc. ACM Manag. Data* 3, 2 (PODS), Article 92 (May 2025), 24 pages. https://doi.org/10.1145/3725229

1 Introduction

Recent work [16] considered the circuit size bounds for the *sum-product polynomial* for a self-join-free Conjunctive Query (CQ). Formally, the sum-product polynomial for a Boolean CQ Q (with self-joins or not) is parameterized by an underlying semiring $\mathbb{S}$ and an instance I

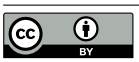
$$p_I^Q := \bigoplus_{h \in \mathrm{Hom}(Q,I)} \bigotimes_{R(\mathbf{x}) \in Q} x_{R(h(\mathbf{x}))} \tag{1}$$

where $\mathrm{Hom}(Q, I)$ is the set of all homomorphisms from Q to I and $x_{R(\mathbf{a})}$ is a variable that annotates the tuple $R(\mathbf{a})$ in I, taking values in the domain of the semiring.

The central question we study in this paper is: *given a CQ (possibly with self-joins), a semiring $\mathbb{S}$, and a set of constraints on an instance I, what is the smallest circuit over $\mathbb{S}$ for CQ on I?*

We briefly introduce the notion of a circuit over a semiring. A circuit F over a semiring $\mathbb{S}$ is a Directed Acyclic Graph (DAG) with input nodes the variables $x_{R(\mathbf{a})}$ of fan-in 0 and every other node labelled by $\oplus$ or $\otimes$ of fan-in 2; these nodes are called $\oplus$-*gates* and $\otimes$-*gates* respectively [22]. A circuit F that computes the sum-product polynomial has an output gate with fan-out 0 that, by sequentially going through all gates in a bottom-up fashion, evaluates the polynomial. The size of the circuit F, denoted as $|F|$, is the number of gates in F. The study of circuits over a semiring is fruitful due to the following two reasons (see also [16] for a more detailed discussion):

Authors' Contact Information: Austen Z. Fan, University of Wisconsin–Madison, Madison, USA, afan@cs.wisc.edu; Paraschos Koutris, University of Wisconsin–Madison, Madison, USA, paris@cs.wisc.edu; Hangdong Zhao, University of Wisconsin–Madison, Madison, USA, hangdong@cs.wisc.edu.



ACM 2836-6573/2025/5-ART92
https://doi.org/10.1145/3725229

(1) Circuits over a semiring can be seen as a model of computation that corresponds to algorithms that *solely* exploit the two algebraic operations in the underlying semiring. This model of computation is equivalent to monotone straight-line programs that computes a function given by a polynomial [20, 40]. In our setting, the polynomial is restricted to be the sum-product polynomial derived from a CQ on an instance I over a semiring $\mathbb{S}$.
(2) Circuits that compute the sum-product polynomial of a CQ can be viewed as a *concise representation* of the corresponding provenance polynomial over a semiring, as introduced introduced by Green, Karvounarakis and Tannen [19]. It has been noted [16] that d-representations in factorised databases [5, 34, 35] can be seen as special cases of circuits for CQs where they are governed by exactly one tree decomposition.

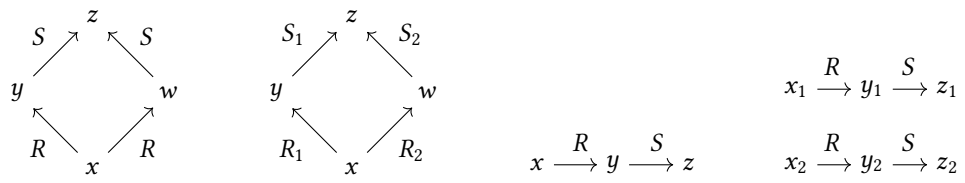
The results in [16] have two limitations: they only work for self-join-free queries, and the lower bounds only apply to semirings that are at least as expressive as the tropical semiring (for instance, no lower bound was proved for the Boolean semiring $\mathbb{B}$). In this work, we take on both challenges.

CQs with self-joins are notoriously difficult to deal with. Many areas, including enumeration complexity [4, 8, 9], consistent query answering [30, 31] and factorised databases [5, 34, 35] enjoy a long line of success when restricted to self-join-free CQs, but immediately run into obstacles when trying to extend the results to CQs with self-joins. To have a slight taste of the difficulty, note that when a full CQ Q is self-join-free, a hypergraph suffices to encode all the information in Q; however, that is not the case for a CQ with self-joins, since different atoms might correspond to the same hyperedge. The fundamental technical difficulty is that the sum-product polynomials for CQs with self-joins are not multilinear and this makes the lower bound approach of [16] fail.

The same technical issue occurs when we consider circuits over the Boolean semiring. In this setting, even for self-join-free CQs, the polynomial produced by the optimal circuit may not be multilinear. Hence, we have to use fundamentally different arguments compared to [16] in order to make progress in the lower bounds.

Our Contributions. We next summarize our contributions in this work.

- In Section 3, we prove for any self-join-free CQ a circuit size a lower bound over $\mathbb{B}$, using the technique of *clique embeddings* [15] and results by Razborov and Alon-Boppana [1, 38] (Theorem 3.4). We show this bound is tight for any binary CQ with subquadratic submodular width using the notion of *full $\mathcal{H}$-encodings* in [7] (Theorem 3.10).
- In Section 4, we present some basic results on CQs with self-joins. Given a CQ Q, we use $[Q]$ to denote the self-join-free CQ which replaces atoms that share the same relational symbol with distinct fresh relational symbols. In a technical sense, the circuit complexity of Q is never higher than that of $[Q]$, since Q can be seen as $[Q]$ while imposing a restriction on input instances. To deal with the opposite direction, we use the technique of *tagging* [8, 9] and deliver a taste of our circuit reductions between self-join-free CQs and CQs with self-joins (Theorem 4.1 & Theorem 4.2).
- In Section 5, we characterize the circuit size of sum-product polynomials for CQs with self-joins over any semiring from the class $\mathbf{C}_{\text{hom}}$ [29], which includes the Boolean semiring and any distributive lattice. For $\mathbf{C}_{\text{hom}}$, the circuit complexity is characterized by its *core* (Theorem 5.7). We prove that several properties of the core of a CQ from $\mathbb{B}$ extend to any semiring in $\mathbf{C}_{\text{hom}}$. We also prove that the image of an endomorphism has decreasing entropic width, a result that might be of independent interest (Theorem 5.9).
- In Section 6, we characterize the circuit size of sum-product polynomials for CQs with self-joins for the semiring class $\mathbf{C}_{=}$ [29] by their self-join-free counterpart (Theorem 6.3). Notable examples in $\mathbf{C}_{=}$ are $\mathbb{T}$, $\mathbb{C}$ and $\mathbb{B}[X]$. To achieve this, we leverage the celebrated result by Baur and Strassen [6] on taking the *partial derivatives of an arithmetic circuit.*



(a) Q_1 (b) $[Q_1]$ (c) Q_2 (d) Q_3

Fig. 1. Some examples of Boolean CQs.

- In Section 7, we characterize the circuit size of sum-product polynomials for CQs with self-joins over the positive tropical semiring $\mathbb{T}^+$ (Theorem 7.8). We define a key object called *tropical core* (Definition 7.6) which captures the circuit complexity of CQ over $\mathbb{T}^+$. The proof is a combination of techniques including geometry in linear programming, endomorphisms of CQs, and extracting polynomials from a circuit by taking partial derivatives.

For ease of reading, we only report results when the set of constraints on input instances consists of uniform cardinality constraints [26]. However, all results in the main text can be easily extended to *degree-aware constraints*. We record these results in Appendix B.

2 Preliminaries

In this section, we present an overview of important terminologies used throughout the paper.

Conjunctive Queries. A Conjunctive Query (CQ) is defined as an expression consisting of a conjunction of atoms:

$$Q(\mathbf{x}_U) \leftarrow R_1(\mathbf{x}_1) \wedge R_2(\mathbf{x}_2) \wedge \cdots \wedge R_\ell(\mathbf{x}_\ell)$$

where $\mathbf{x}_U, \mathbf{x}_i$ are tuples of variables or constants. The variables in $\mathbf{x}_U$ are called *free variables*. We say that Q is *Boolean* if $\mathbf{x}_U$ is the empty tuple and *full* if the free variables are all the variables in the query body. The set of variables of Q is denoted as $\mathrm{var}(Q)$. A CQ is said to have *self-joins* if it contains two atoms with the same relational symbol; otherwise, it is called *self-join-free*. A CQ with self-joins can have an atom occur multiple times, e.g. the CQ $Q \leftarrow R(x,y) \wedge R(x,y)$ is allowed. This means that Q can be viewed as a bag of atoms. Given an instance I, a *homomorphism* from Q to I is a function $h : \mathrm{var}(Q) \to adom(I)$[1] such that for every atom $R_i(\mathbf{x}_i)$ in Q, it holds that $R_i(h(\mathbf{x}_i)) \in I$. We denote by $\mathrm{Hom}(Q, I)$ the set of all homomorphisms from Q to I.

Given a CQ Q, we call the *self-join-free counterpart of* Q, denoted as $[Q]$, to be the CQ which replaces atoms that share the same relational symbol with distinct fresh relational symbols. For example, consider the CQ $Q_1() \leftarrow R(x,y) \wedge R(x,w) \wedge S(y,z) \wedge S(w,z)$ depicted in Figure 1a. Its self-join-free counterpart $[Q_1]() \leftarrow R_1(x,y) \wedge R_2(x,w) \wedge S_1(y,z) \wedge S_2(w,z)$ is depicted in Figure 1b.

With each self-join-free CQ Q we can associate a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where each node represents a variable in Q and each hyperedge contains the variables of an atom in Q.

Semirings. A (commutative) *semiring* is an algebraic structure $\mathbb{S} := (\mathbf{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, where $\oplus$ and $\otimes$ are the *addition* and *multiplication* in $\mathbb{S}$ such that: (1) $(\mathbf{D}, \oplus, \mathbf{0})$ and $(\mathbf{D}, \otimes, \mathbf{1})$ are commutative monoids, (2) $\otimes$ is distributive over $\oplus$, and (3) $\mathbf{0}$ is an annihilator of $\otimes$ in $\mathbf{D}$. We say that $\mathbb{S}$ is *idempotent* if $x \oplus x = x$ for every $x \in \mathbf{D}$ and *absorptive* if $\mathbf{1} \oplus x = \mathbf{1}$ for every $x \in \mathbf{D}$.

Green et al. [19] observed that the semantics of CQ are in natural correspondence to the underlying semiring. For example, set semantics correspond to computing a query over the *Boolean semiring* $\mathbb{B} := (\{0,1\}, \vee, \wedge, \textsc{false}, \textsc{true})$, while bag semantics correspond to the *Counting semiring*

[1] $adom(I)$ is the active domain of an instance I and containts all constants occurring in I.

$\mathbb{C} := (\mathbb{N}, +, \cdot, 0, 1)$ where $\mathbb{N}$ is the set of natural numbers. Other important semirings include the *Tropical semiring* $\mathbb{T} := (\mathbb{Z} \cup \{+\infty\}, \min, +, +\infty, 0)$ which is isomorphic to the *Max semiring* $\mathbb{M} := \{\mathbb{Z} \cup \{-\infty\}, \max, +, -\infty, 0\}$ [2], the *positive tropical semiring* $\mathbb{T}^+ := (\mathbb{R}^+ \cup \{+\infty\}, \min, +, +\infty, 0)$ and the *positive Max semiring* $\mathbb{M}^+ := (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$, all of which are closely related to optimization problems [24, 25]. Our work follows the approach in [19], where an instance is interpreted as an instance tagged with annotations from a semiring (i.e., a K-relation).

Sum-Product Polynomials. The sum-product polynomial for a Boolean CQ Q is parameterized by an underlying semiring $\mathbb{S}$ and an instance I in the following way. For every tuple $R(\mathbf{a})$ in I, let $x_{R(\mathbf{a})}$ be a variable that takes values in the domain of $\mathbb{S}$. Then, the sum-product polynomial for Q of I over $\mathbb{S}$ is defined as

$$p_I^Q := \bigoplus_{h \in \mathrm{Hom}(Q,I)} \bigotimes_{R(\mathbf{x}) \in Q} x_{R(h(\mathbf{x}))} \tag{2}$$

Example 2.1. For Q_1 in Fig. 1 with instance $I = \{R(a,b), R(a,c), S(b,d), S(c,d)\}$ over $\mathbb{T}$, the sum-product polynomial is

$$\begin{aligned} p_I^Q = \min\{&x_{R(a,b)} \oplus x_{R(a,c)} \oplus x_{S(b,d)} \oplus x_{S(c,d)}, \\ &x_{R(a,c)} \oplus x_{R(a,b)} \oplus x_{S(c,d)} \oplus x_{S(b,d)}, \\ &x_{R(a,b)} \oplus x_{R(a,b)} \oplus x_{S(b,d)} \oplus x_{S(b,d)}, \\ &x_{R(a,c)} \oplus x_{R(a,c)} \oplus x_{S(c,d)} \oplus x_{S(c,d)}\} \end{aligned}$$

where $\mathrm{Hom}(Q,I) = \{h_1 : \{x \mapsto a, y \mapsto b, w \mapsto c, z \mapsto d\}, h_2 : \{x \mapsto a, y \mapsto c, w \mapsto b, z \mapsto d\}, h_3 : \{x \mapsto a, y \mapsto b, w \mapsto b, z \mapsto d\}, h_4 : \{x \mapsto a, y \mapsto c, w \mapsto c, z \mapsto d\}\}$.

A sum-product polynomial is always *homogeneous*: the sum of the exponents in every monomial is the same. We also say that it is *multilinear* if the exponent of every variable in a monomial is one. A crucial observation is that if Q is self-join-free, then p_I^Q must be multilinear; however, this is not true if the query has self-joins. We differentiate the notions of *term* and *monomial* for a sum-product polynomial carefully: a term is a formal product of variables, where the product $\otimes$ is interpreted over a semiring, whereas a monomial is a term with a coefficient $a \in \mathbb{N}$, which is interpreted as the number of additions $\oplus$ of the term over the same semiring.

Any sum-product polynomial p over a semiring $\mathbb{S}$ with n variables defines a function $\hat{p} : \mathbb{S}^n \to \mathbb{S}$ whose value $p(s_1, s_2, \ldots, s_n)$ is obtained by substituting elements $s_i \in \mathbb{S}$ for x_i in p. Two sum-product polynomials p_1, p_2 are *equivalent over* $\mathbb{S}$, denoted as $p_1 \equiv_{\mathbb{S}} p_2$, if $\hat{p_1}(\mathbf{s}) = \hat{p_2}(\mathbf{s})$ for all $\mathbf{s} \in \mathbb{S}^n$ [22].

Tree Decompositions. We recall the notion of *tree decomposition* [39].

Definition 2.2 (Tree Decomposition). A *tree decomposition* of a hypergraph $\mathcal{H}$ is a pair $(\mathcal{T}, \chi)$, where $\mathcal{T}$ is a tree and χ maps each node t of the tree to a subset $\chi(t)$ of $V(\mathcal{H})$, called a bag, s.t.:

(1) every hyperedge $e \in E(\mathcal{H})$ is a subset of $\chi(t)$ for some $t \in V(\mathcal{T})$; and
(2) for every vertex $v \in V(\mathcal{H})$, the set $\{t | v \in \chi(t)\}$ is a non-empty connected subtree of $\mathcal{T}$.

We say that a tree decomposition is *non-redundant* if no bag is a subset of another; otherwise it is *redundant*. We let $\mathrm{TD}(\mathcal{H})$ be the set of all non-redundant tree decompositions of $\mathcal{H}$.

Entropic and Submodular Width. Let n be a positive integer. A function $h : 2^{[n]} \to \mathbb{R}_+$ is called a *set function* on $[n] = \{1, 2, \ldots, n\}$. A set function is an *entropic function* of order n if there exist random variables $A_1, \ldots, A_n$ such that $h(S) = H((A_i)_{i \in S})$ for any $S \subseteq [n]$, where H is the joint

[2]The additional element $-\infty$ to $\mathbb{Z}$ is to make sure that the additive identity and the multiplicative identity are different, and also that the additive identity is an annihilator of $\otimes := +$ in $\mathbb{M}$.

entropy of a set of variables. We denote by Γ_n^* the set of all entropic functions of order n, $\overline{\Gamma}_n^*$ the topological closure of Γ_n^* and Γ_n the submodular set functions on $[n]$ (Definition 2.2 in [27]). For a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $|\mathcal{V}| = n$, define:

$$\mathsf{ED} := \{h : 2^{[n]} \to \mathbb{R}_+ \mid h(e) \leq 1, \forall e \in \mathcal{E}\}.$$

Then, the *fractional hypertree width*, *submodular width* and *entropic width* are defined as follows (see also [27]):

$$\mathsf{fht}(\mathcal{H}) := \min_{(\mathcal{T},\chi)\in\mathsf{TD}} \max_{h\in\Gamma_n\cap\mathsf{ED}} \max_{v\in V(\mathcal{T})} h(\chi(v))$$

$$\mathsf{subw}(\mathcal{H}) := \max_{h\in\Gamma_n\cap\mathsf{ED}} \min_{(\mathcal{T},\chi)\in\mathsf{TD}} \max_{v\in V(\mathcal{T})} h(\chi(v))$$

$$\mathsf{entw}(\mathcal{H}) := \max_{h\in\overline{\Gamma}_n^*\cap\mathsf{ED}} \min_{(\mathcal{T},\chi)\in\mathsf{TD}} \max_{v\in V(\mathcal{T})} h(\chi(v))$$

Circuits over Semirings. A *circuit* F over a semiring $\mathbb{S}$ is a Directed Acyclic Graph (DAG) with input nodes (with fan-in 0) variables in a set S_x containing $x_{R(\mathbf{a})}$'s in the RHS of Equation 2 and the constants $\mathbf{0}, \mathbf{1}$. Every other node is labelled by $\oplus$ or $\otimes$ and has fan-in 2; these nodes are called $\oplus$-*gates* and $\otimes$-*gates*, respectively. An input gate of F is any gate with fan-in 0 and an output gate of F is any gate with fan-out 0. The size of the circuit F, denoted as $|F|$, is the number of gates in F. The depth of F is the length of the longest path from an input to an output node.

In this paper, the goal of circuits over some semiring is to compute sum-product polynomials over that semiring. A circuit F can be viewed as a concise representation of a polynomial $\hat{F}$ over the variable set S_x. A circuit F is said to *compute* a polynomial p if $\hat{F}$ and p coincide as functions (interpreted over the semiring $\mathbb{S}$), and is said to *produce* a polynomial p if $\hat{F}$ and p, when written as sums of products of variables and coefficients, are exactly the same modulo the ordering of the summands and the ordering of variables and coefficients within each summand.

A circuit F is *homogeneous* if every polynomial produced in each gate in F is homogeneous, and is *multilinear* if for every $\otimes$-gate $u = v \otimes w$, the polynomials produced at gates v and w have a disjoint set of variables.

Let $\mathbb{S}(p)$ and $\mathbb{S}[p]$ be the minimum size of a circuit over semiring $\mathbb{S}$ *computing* and *producing* p, respectively. If F produces p, then F must compute p, but not necessarily the other way around.

We also define $\mathbb{S}(Q, N)$ for some integer $N > 0$ to be the size of the smallest circuit computing the polynomial p_I^Q over $\mathbb{S}$ over all instances I that have relations with size at most N. In other words, $\mathbb{S}(Q, N)$ is a worst-case bound. The following tight circuit bounds were proved in [16].

Theorem 2.3 (Lower Bound [16]). *For any self-join-free CQ Q with hypergraph $\mathcal{H}$, and any semiring $\mathbb{S} \in \{\mathbb{B}_{\mathsf{lin}}, \mathbb{T}, \mathbb{C}\}$* [3] *it holds that $\mathbb{S}(Q, N) = \Omega(N^{\mathsf{entw}(\mathcal{H})})$.*

Theorem 2.4 (Upper Bound [16]). *Fix a self-join-free CQ Q. Let I be an instance where every relation has size at most N. Then, there exists a multilinear and homogeneous circuit F of size $O(N^{\mathsf{entw}(\mathcal{H})})$ that computes the polynomial p_I^Q over any idempotent semiring.*

When proving the above results, the authors of [16] applied the following lemma to transfer a circuit size lower bound from one semiring to another.

Lemma 2.5 ([22]). *If p is a multilinear and homogeneous polynomial, then the following equalities hold: $\mathbb{B}_{\mathsf{lin}}(p) = \mathbb{T}(p) = \mathbb{T}[p] = \mathbb{C}(p) = \mathbb{C}[p]$.*

[3] We slightly abuse the notation; strictly speaking, $\mathbb{B}_{\mathsf{lin}}$ is not a semiring but a requirement on the circuit computing over $\mathbb{B}$. Formally, $\mathbb{B}_{\mathsf{lin}}(p)$ is the minimum size of a *multilinear* circuit over $\mathbb{B}$ computing p.

Semiring Homomorphisms. We define here the notion of a semiring homomorphism, which implies a hierarchy of semirings that allows us to transfer upper and lower bounds.

Definition 2.6 (Semiring Homomorphism). Let $\mathbb{S}_1, \mathbb{S}_2$ be two semirings. A mapping $h : \mathbb{S}_1 \to \mathbb{S}_2$ is called a semiring homomorphism if $h(\mathbf{0}) = \mathbf{0}, h(\mathbf{1}) = \mathbf{1}$, and for all $a, b \in \mathbb{S}_1$, we have $h(a \oplus b) = h(a) \oplus h(b)$ and $h(a \otimes b) = h(a) \otimes h(b)$.

Lemma 2.7. *If there exists a surjective semiring homomorphism $h : \mathbb{S}_1 \to \mathbb{S}_2$, then $\mathbb{S}_1(p) \geq \mathbb{S}_2(p)$ for any polynomial p.*

Proof. We will show that if F is a circuit that computes a polynomial p over semiring $\mathbb{S}_1$, then F computes p over semiring $\mathbb{S}_2$ as well.

Let p_F be the polynomial produced by the circuit F. Then, it must be that $p_F \equiv_{\mathbb{S}_1} p$, meaning $\hat{p_F}(\mathbf{s_1}) = \hat{p}(\mathbf{s_1})$ for any $\mathbf{s_1} \in \mathbb{S}_1^n$. Now, for any $\mathbf{s_2} \in \mathbb{S}_2^n$, let $\mathbf{s_1}$ be any vector in $\mathbb{S}_1^n$ such that $h(\mathbf{s_1}) = \mathbf{s_2}$, where h applies component-wise. Then, observe that

$$\hat{p_F}(\mathbf{s_2}) = \hat{p_F}(h(\mathbf{s_1})) = h(\hat{p_F}(\mathbf{s_1})) = h(\hat{p}(\mathbf{s_1})) = \hat{p}(h(\mathbf{s_1})) = \hat{p}(\mathbf{s_2})$$

where the second and fourth equality comes from the definition of a semiring homomorphism. Since $\mathbf{s_2}$ is arbitrary, we conclude that $p_F \equiv_{\mathbb{S}_2} p$. □

We call a semiring $\mathbb{S}$ *positive* if (i) $\mathbf{0} \neq \mathbf{1}$, (ii) $a \oplus b = \mathbf{0}$ implies $a = \mathbf{0}$ and $b = \mathbf{0}$, and (iii) $a \otimes b = \mathbf{0}$ implies $a = \mathbf{0}$ or $b = \mathbf{0}$. If $\mathbb{S}$ is positive, then there exists a (surjective) semiring homomorphism to the Boolean semiring $\mathbb{B}$ [18] . Hence by Lemma 2.7, any lower bound for the Boolean semiring transfers to all positive semirings (which include all semirings we consider in the paper).

Homomorphisms, Endomorphisms, Automorphisms. A mapping h from the variables of a CQ Q_1 to the variables of a CQ Q_2 is a *homomorphism from Q_1 to Q_2* if (1) $R(h(\mathbf{x})) \in Q_2$ whenever $R(\mathbf{x}) \in Q_1$ and (2) $h(\mathbf{x}_{U_1}) = h(\mathbf{x}_{U_2})$ where $\mathbf{x}_{U_1}$ and $\mathbf{x}_{U_2}$ are the free variables of Q_1 and Q_2 respectively. An *endomorphism of Q* is a homomorphism from Q to itself. An *automorphism of Q* is an endomorphism of Q that is also an isomorphism.

Example 2.8. Consider the query Q_1 from Figure 1. The mapping $x \to x, y \to w, w \to y, z \to z$ is an automorphism for Q_1, while the mapping $x \to x, y \to y, w \to y, z \to z$ is an endomorphism but not an automorphism.

CQ Equivalence. Given two CQs Q_1 and Q_2, we say they are *syntactically equivalent* if they are identical up to variable renaming. This notion is also called *(body) isomorphism* in [8, 12, 13]. For example, the CQs $Q_1() \leftarrow R(x) \wedge S(x, y)$ and $Q_2() \leftarrow S(y, x) \wedge R(y)$ are syntactically equivalent. In contrast, we say that two queries are semantically equivalent over a semiring $\mathbb{S}$, denoted as $Q_1 \equiv_{\mathbb{S}} Q_2$, if upon any $\mathbb{S}$-annotated input relation I, Q_1 and Q_2 return the same result.

For an endomorphism h, by $h(Q)$ we mean the query that contains all variables and atoms that are in the image of h. In other words, h is surjective when viewed *both* as mappings from Q to $h(Q)$ between variables and atoms. The following lemma gives a characterization when two queries are syntactically equivalent in terms of homomorphisms (since the converse of the lemma is obvious).

Lemma 2.9. *If there exists a homomorphism h from Q_1 to Q_2 and a homomorphism g from Q_2 to Q_1 such that $h(Q_1) = Q_2$ and $g(Q_2) = Q_1$, then Q_1 and Q_2 are syntactically equivalent.*

Proof. Clearly, h and g are bijective when viewed as mappings between variables. Since $g \circ h(Q_1) = Q_1$ and $g \circ h(Q_2) = Q_2$, the mappings between atoms in Q_1 and Q_2 that are induced by h and g must be bijective as well. This implies that Q_1 and Q_2 are the same up to variable renaming, as witnessed by h and g. □

Homomorphic Equivalence. Two CQs Q_1 and Q_2 are said to be *homomorphically equivalent* if there exists a homomorphism h from Q_1 to Q_2 and a homomorphism g from Q_2 to Q_1. This is a syntactic property between two CQs, and has been leveraged heavily in several prior works [3, 12, 14, 41]. It is not hard to show that for a large class of semirings, two homomorphically equivalent queries are semantically equivalent over any semiring in the class. In fact, Theorem 5.4 shows that over the class of semirings called $\mathbf{C}_{\text{hom}}$ defined in Section 5, two CQs Q_1 and Q_2 are homomorphically equivalent if and only if they are semantically equivalent. The situation could be different outside $\mathbf{C}_{\text{hom}}$. Consider the queries Q_1, Q_2 and Q_3 in Figure 1. Note that all these queries are homomorphically equivalent to each other. However, Q_1 and Q_2 are not semantically equivalent over $\mathbb{T} \notin \mathbf{C}_{\text{hom}}$, since Q_1 will always have 2 times the result of Q_2. Moreover, Q_1 and Q_3 are semantically equivalent over $\mathbb{T}$ but not semantically equivalent over $\mathbb{C} \notin \mathbf{C}_{\text{hom}}$. Finally, even over $\mathbb{B} \in \mathbf{C}_{\text{hom}}$, in which case $Q_1 \equiv_{\mathbb{B}} Q_2 \equiv_{\mathbb{B}} Q_3$, we note that Q_1, Q_2 and Q_3 are not syntactically equivalent.

3 Lower Bounds for Boolean circuits for self-join-free CQs

In this section, all CQs are assumed to be self-join-free, so we will identify the CQ with its hypergraph $\mathcal{H}$. Our goal is to prove lower bounds on the size of circuits over the Boolean semiring. This is quite challenging, since the techniques developed in [16] to show tight lower bounds cannot be applied to the Boolean semiring. The main technical issue is that a Boolean circuit that computes a multilinear polynomial (as is $p_I^{\mathcal{H}}$) does not need to be multilinear, and the arguments in [16] only apply for multilinear circuits.

3.1 Lower Bounds via Clique Embeddings

Fortunately, we have a starting point, which is a classical result by Razborov [38], later improved by Alon and Boppana [1]. To state the result, define the polynomial C_k^n over $\binom{n}{2}$ Boolean variables that takes as input an undirected graph on n vertices (where each variable $x_{\{i,j\}}$ corresponds to whether the edge $\{i, j\}$ exists for $1 \le i \ne j \le n$) and returns true iff the graph contains a k-clique.

Theorem 3.1 (Alon & Boppana [1]). *For any constant $k \ge 3$, every circuit F over the Boolean semiring that computes C_k^n must have size $|F| = \Omega\left(\frac{n^k}{\log^k n}\right)$.*

In particular, we cannot have a Boolean circuit of size $O(n^{k-\epsilon})$ for some constant $\epsilon > 0$. To be able to apply the above result to our setting, we need two steps. The first step is to show that the lower bound works for the k-clique polynomial as defined over a k-partite graph. Concretely, we define the polynomial P_k^n as follows. The input is a k-partite graph with vertex set $V_1 \times \cdots \times V_k$, each V_i of size n (this means no edge exists within a partition V_i). The output is true if and only if the graph contains a k-clique; such a clique must have exactly one vertex in each partition V_i. It is clear that given a Boolean circuit F that computes C_k^n, one can construct a Boolean circuit F that computes P_k^n. We show that the converse is also true.

Lemma 3.2. *For any constant $\epsilon > 0$ and $k \ge 3$, there exists no circuit that computes P_k^n over the Boolean semiring with size $O(n^{k-\epsilon})$.*

Proof of Lemma 3.2. It suffices to show that given a circuit over $\mathbb{B}$ that computes the polynomial P_k^n, one can construct a circuit over $\mathbb{B}$ that computes C_k^n. Let F be a circuit over the Boolean semiring that computes P_k^n. The input gates of the circuit are of the form $y_{\{(i,p),(j,q)\}}$ where $1 \le i, j, \le n$ and $1 \le p \ne q \le k$: this is the gate that corresponds to the edge that connects vertex i of partition V_p to vertex j of partition V_q. We construct a circuit F' for C_k^n as follows. It is identical to F, except for the input wires which we reroute to the new input gates. In particular, if the input gate is $y_{\{(i,p),(j,q)\}}$ and $i = j$ we reroute it to $\mathbf{0}$, otherwise we reroute it to the input gate $x_{\{i,j\}}$. It is easy to see that F'

correctly computes C_k^n, since we have the property that every k-clique formed in P_k^n needs to have exactly one vertex on each partition (and of course all vertices are distinct). □

For the second step, we need to apply the idea of a *clique embedding* [15]. Intuitively, a clique embedding embeds an instance of a clique problem inside a query, so that an algorithm that solves the query will also solve the clique problem. This idea was used to prove conditional lower bounds for query evaluation [15], but here we will use it to get unconditional lower bounds for circuits.

We will first need some definitions. We say that two sets of vertices $X, Y \subseteq V(H)$ *touch* in $\mathcal{H}$ if either $X \cap Y \neq \emptyset$ or there is a hyperedge $e \in E(\mathcal{H})$ that intersects both X and Y.

Definition 3.3 (Clique Embedding). Let $k \geq 3$ and $\mathcal{H}$ be a hypergraph. A *k-clique embedding*, denoted $C_k \mapsto H$, is a mapping ψ that maps every $v \in \{1, \ldots, k\}$ to a non-empty subset $\psi(v) \subseteq V(\mathcal{H})$ such that the following hold:

(1) $\psi(v)$ induces a connected subgraph;
(2) for any two $u \neq v \in \{1, \ldots, k\}$ then $\psi(u), \psi(v)$ *touch* in $\mathcal{H}$.

It will often be convenient to describe a clique embedding ψ by the reverse mapping $\psi^{-1}(x) = \{i \mid x \in \psi(i)\}$, where x is a vertex in $V(\mathcal{H})$. For a hyperedge $e \in E(\mathcal{H})$, its *weak edge depth* is $d_\psi(e) = |\{v \mid \psi(v) \cap e \neq \emptyset\}|$, i.e. the number of colors from $[k]$ that map to some vertex in e. The *weak edge depth* of an embedding ψ can then be defined as $\mathsf{wed}(\psi) = \max_e d_\psi(e)$. Additionally, we define as $\mathsf{wed}(C_k \mapsto H)$ the minimum weak edge depth of any clique embedding ψ. We can now define the following quantity, which we call the *clique embedding power*:

$$\mathsf{clemb}(\mathcal{H}) := \sup_{k \geq 3} \frac{k}{\mathsf{wed}(C_k \mapsto \mathcal{H})}.$$

It has been shown [15] that this quantity is well-defined. As an example, the CQ that corresponds to an cycle of length ℓ has $\mathsf{clemb} = 2 - 1/\lceil \ell/2 \rceil$.

Equipped with this definition, we will now show the main theorem of this section.

Theorem 3.4. *For a self-join-free CQ Q with hypergraph $\mathcal{H}$ and any constant $\epsilon > 0$, we have that $\mathbb{B}(Q, N) = \Omega(N^{\mathsf{clemb}(\mathcal{H}) - \epsilon})$.*

Proof. From [15], we know that there exists some $k > 0$ such that $\mathsf{clemb}(\mathcal{H}) = \frac{k}{\mathsf{wed}(C_k \mapsto \mathcal{H})}$. Let ψ be the k-clique embedding from C_k to $\mathcal{H}$ that achieves the clique embedding power for $\mathcal{H}$, with weak edge depth $\lambda = \frac{k}{\mathsf{clemb}(\mathcal{H})}$.

To prove the statement, we will reduce the k-partite k-clique problem to the problem of computing the query $\mathcal{H}$, but our reduction will be through circuits. The first step is to construct an input instance I for $\mathcal{H}$. We will do this following the same logic as [15]. For any variable $x \in V(\mathcal{H})$, define the domain $\mathsf{Dom}(x)$ in I as vectors over $[n]^{|\psi^{-1}(x)|}$. Let $S_e = \{i \in [k] \mid \psi(i) \cap e \neq \emptyset\}$. We have $|S_e| \leq \lambda$ by the definition of weak edge depth. The idea is that each input tuple for hyperedge e corresponds to a clique between the partitions V_i where $i \in S_e$; note that these cliques will have size $|S_e| \leq \lambda$ and there are at most $O(n^\lambda)$ such cliques. Formally, for every $\mathbf{a} \in [n]^{S_e}$, we introduce a tuple $t_\mathbf{a}$ in the instance I over $\prod_{i \in S_e} \mathsf{Dom}(x_i)$ whose value at variable x is $\langle \mathbf{a_i} \mid i \in \psi^{-1}(x) \rangle$.

Now, consider the circuit F that computes the polynomial $p_I^{\mathcal{H}}$ over the Boolean semiring. We will construct a circuit F' for P_k^n as follows. F' can be seen as composition of two parts. The first part takes as input the input variables $y_{\{(i,p),(j,q)\}}$ of P_k^n. Then, for every hyperedge e and $\mathbf{a} \in [n]^{S_e}$, it constructs an output gate that computes the "value" of the tuple $t_\mathbf{a}$: in particular, it computes $\bigwedge_{p,q \in S_e, p \neq q} y_{\{(\mathbf{a}_p,p),(\mathbf{a}_q,q)\}}$. In other words, the gate for $t_\mathbf{a}$ will be true if and only if there is an S_e-clique. The second part is identical to F but connects the input wires to the corresponding output

gate of the tuple computed in the first part. By the same argument as in [15], we can argue that F' correctly computes the k-clique k-partite problem, and so the desired polynomial.

We now argue about the size of the circuit. From our instance construction, $|I| = O(n^\lambda)$. This implies that the size of the first part of F' is $O(n^\lambda)$. The second part of the circuit has size $|F|$. Now, suppose for the sake of contradiction that $\mathbb{B}(\mathcal{H}, N) = O(N^{\mathsf{clemb}(\mathcal{H})-\epsilon})$ for some $\epsilon > 0$. Then, $|F| = O(|I|^{\mathsf{clemb}(\mathcal{H})-\epsilon})$. Hence, the resulting circuit F for P_k^n has size $O(n^\lambda + (n^\lambda)^{\mathsf{clemb}(\mathcal{H})-\epsilon}) = O(n^{k-\epsilon'})$ where $\epsilon' = \lambda\epsilon$. This contradicts Lemma 3.2 and thus completes the proof. □

Using semiring homomorphisms and Lemma 2.7, we obtain as a corollary the following lower bound for all positive semirings:

Corollary 3.5. *Let $\mathbb{S}$ be a positive semiring. For a self-join-free CQ with hypergraph $\mathcal{H}$ and any constant $\epsilon > 0$, we have that $\mathbb{S}(Q, N) = \Omega(N^{\mathsf{clemb}(\mathcal{H})-\epsilon})$.*

Since the Boolean semiring is idempotent, by applying Theorem 2.4 we also get an upper bound of $\mathbb{B}(Q, N) = O(N^{\mathsf{entw}(\mathcal{H})})$ for the Boolean semiring. This seems to imply that the clique embedding power is bounded by entropic width. We next give a direct proof of this fact that may be of independent interest, since the previously best upper bound known for clemb is that $\mathsf{clemb}(\mathcal{H}) \leq \mathsf{subw}(\mathcal{H})$.

Theorem 3.6. *For any hypergraph $\mathcal{H}$, we have* $\mathsf{clemb}(\mathcal{H}) \leq \mathsf{entw}(\mathcal{H})$.

Proof. Let $\mathsf{wed}(C_k \to \mathcal{H}) = \alpha$. Then, there is an embedding $\psi : C_k \to \mathcal{H}$ with weak edge depth α. We will show that $\mathsf{entw}(\mathcal{H}) \geq k/\alpha$.

To show this, we will need to define a vector $h \in \overline{\Gamma}_n^*$ over subsets of $V(\mathcal{H})$ that is edge-dominated. To do this, we construct an instance I as follows. For convenience, assume $V(C_k) = \{1, 2, \ldots, k\}$; we call these vertices colors. For every variable $x \in V(\mathcal{H})$, let $\psi^{-1}(x)$ be the subset of $\{1, \ldots, k\}$ mapping to x. Then, we define the domain of variable x, $\mathsf{Dom}(x)$ to be $\{0,1\}^{|\psi^{-1}(x)|}$. Now, for every vector $\mathbf{b} \in \{0,1\}^k$, we introduce one tuple $t_e^{\mathbf{b}}$ in every relation R_e such that $t_e^{\mathbf{b}}[x] = \langle \mathbf{b}_i \mid i \in \psi^{-1}(x) \rangle$.

It is easy to see that the output of the query over instance I is exactly 2^k tuples. We define now a joint probability distribution over the output tuples by choosing each output tuple with the same uniform probability. The entropy of this distribution has $\bar{h}([n]) = \log 2^k = k$.

Now, consider any decomposition $(\mathcal{T}, \chi)$ of $\mathcal{H}$. From Lemma 6 of [15], there is a node $t \in V(\mathcal{T})$ such that or every color $i \in \{1, \ldots, k\}$, $\psi(i) \cap \chi(t) \neq \emptyset$. This implies that the projections of the 2^k output tuples on $\chi(t)$ are pairwise disjoint, and hence $\bar{h}(\chi(t)) = \bar{h}([n]) = k$. Hence, for any decomposition $(\mathcal{T}, \chi)$, $\max_{v \in V(\mathcal{T})} \bar{h}(\chi(v)) = k$.

Finally, we define $h = \bar{h}/\alpha$. Since $\bar{h}$ is entropic, $h \in \overline{\Gamma}_n^*$. We claim that h is also edge-dominated. Indeed, since the weak edge depth is α, each hyperedge e contains at most α colors, and hence $\bar{h}(e) \leq \log 2^\alpha = \alpha$, which implies that $h(e) \leq 1$. Finally,

$$\min_{(\mathcal{T},\chi) \in \mathsf{TD}} \max_{v \in V(\mathcal{T})} h(\chi(v)) = k/\alpha$$

which implies that $\mathsf{entw}(\mathcal{H}) \geq k/\alpha$. □

3.2 Tight Bounds

In this part, we show classes of self-join-free CQs for which we can obtain tight upper and lower bounds for any positive semiring (which essentially includes all semirings of interest). The first class of queries we consider are chordal queries.

Chordal Queries. A *chord* of a cycle C of a graph G is an edge that connects any two non-adjacent vertices in C. We say that G is a *chordal graph* if any cycle of length greater than 3 has a chord. We say that a hypergraph $\mathcal{H}$ is *chordal* if its clique-graph is chordal, where the clique graph adds an

edge between all pairs of vertices contained in the same hyperedge. Examples of chordal queries include the Loomis-Whitney join, all acyclic queries, and clique queries.

Proposition 3.7. *Let Q be a self-join-free CQ with a chordal hypergraph $\mathcal{H}$, and $\mathbb{S}$ be a positive semiring. Then, $\mathbb{S}(Q,N) = O(N^{\mathsf{clemb}(\mathcal{H})})$, but for any $\epsilon > 0$, $\mathbb{S}(Q,N) = \Omega(N^{\mathsf{clemb}(\mathcal{H})-\epsilon})$.*

Proof. It was shown in [16] that $O(N^{\mathsf{fhw}(\mathcal{H})})$ is an upper bound for the circuit size over any semiring $\mathbb{S}$ where $\mathsf{fhw}(\mathcal{H})$ is the fractional hypertree width of $\mathcal{H}$. Moreover, for any $\epsilon > 0$, $\Omega(N^{\mathsf{clemb}(\mathcal{H})-\epsilon})$ is a lower bound for the size of any Boolean semiring circuit. Since $\mathbb{S}$ is positive, this lower bound transfers to the semiring $\mathbb{S}$. Finally, because $\mathcal{H}$ is chordal, it holds that $\mathsf{clemb}(\mathcal{H}) = \mathsf{fhw}(\mathcal{H})$ [15]. This implies that we have a tight upper and lower bound for any positive semiring. □

Queries with Subquadratic Submodular Width. Let $\mathcal{H}$ be a hypergraph such that each relation is binary ($\mathcal{H}$ is a graph). In [7], the authors identified a class of graphs (precisely the graphs with $\mathsf{subw}(\mathcal{H}) < 2$) such that the corresponding CQ can be computed in time $O(N^{\mathsf{clemb}(\mathcal{H})})$. We will show how to turn this result into a circuit upper bound for any semiring. For this, we need to define the notion of a full $\mathcal{H}$-encoding, as in [7].

Definition 3.8 (Full $\mathcal{H}$-encoding). Let $\mathcal{H}$ be the hypergraph of a self-join-free CQ Q, and I be an instance. A *partial $\mathcal{H}$-encoding* $\mathcal{E}$ is a tree decomposition $(\mathcal{T}, \chi)$ of $\mathcal{H}$ together with a collection of sets $\{S_v\}_{v \in V(\mathcal{T})}$ of tuples. We say that a tuple $t \in Q(I)$ is encoded by $\mathcal{E}$ if $t[\chi(v)] \in S_v$ for every $v \in V(\mathcal{T})$. A *full $\mathcal{H}$-encoding* is a collection $\{\mathcal{E}_i\}_i$ of partial $\mathcal{H}$-encodings such that each tuple $t \in Q(I)$ is encoded by exactly one $\mathcal{E}_i$. Moreover, the number of sets in the collection is finite and depends only on $\mathcal{H}$ [4].

Intuitively, if we have an $\mathcal{H}$-encoding, we have a collection of tree decompositions where each bag is materialized. It is critical that an output tuple is encoded by exactly one partial $\mathcal{H}$-encoding. The size of a full $\mathcal{H}$-encoding is defined as the sum of the sizes of all sets S_v.

Proposition 3.9. *Suppose that there exists a full $\mathcal{H}$-encoding of an instance I of size S. Then, we can construct a circuit of size $O(S)$ that computes $p_I^{\mathcal{H}}$ over any positive semiring.*

Proof. We follow the same construction as in [16]. The key difference is that the correctness does not depend on the idempotency of the semiring, since by the definition of a full $\mathcal{H}$-encoding we are guaranteed that each tuple is produced exactly once across the circuit. □

Note that the circuit we construct works over any positive semiring, even non-idempotent ones. We can show the following result.

Theorem 3.10. *Let Q be a self-join-free CQ with a hypergraph $\mathcal{H}$ with binary relations and $\mathsf{subw}(\mathcal{H}) < 2$. Let $\mathbb{S}$ be a positive semiring. Then, $\mathbb{S}(Q,N) = O(N^{\mathsf{clemb}(\mathcal{H})})$, but for any $\epsilon > 0$, $\mathbb{S}(Q,N) = \Omega(N^{\mathsf{clemb}(\mathcal{H})-\epsilon})$.*

Proof. The lower bound of $\Omega(N^{\mathsf{clemb}(\mathcal{H})})$ is obtained as before. For the upper bound, we use the result in [7] that shows that when $\mathsf{subw}(\mathcal{H}) < 2$ and $\mathcal{H}$ is binary, we can construct an $\mathcal{H}$-encoding of size $\Omega(N^{\mathsf{clemb}(\mathcal{H})})$. From 3.9, this implies a circuit of the same size that computes the desired polynomial over every semiring. □

Note that for this class of queries, we also have that $\mathsf{clemb}(\mathcal{H}) = \mathsf{entw}(\mathcal{H}) = \mathsf{subw}(\mathcal{H})$, since the entropic width is sandwiched between the clique embedding power and the submodular width. Examples of queries in this class consist of all cycles, as well as all parallel-series graphs. We record this observation as the following theorem.

[4] The total number of elements in all the sets of the collection will generally not be finite and depends on I.

Theorem 3.11. *Let Q be a self-join-free CQ with a hypergraph $\mathcal{H}$ with binary relations and* $\mathsf{subw}(\mathcal{H}) < 2$. *Then,* $\mathsf{clemb}(\mathcal{H}) = \mathsf{entw}(\mathcal{H}) = \mathsf{subw}(\mathcal{H})$.

4 CQs with self-joins: Some Basic Results

We now switch our attention to CQs with self-joins and present some basic results. Since the polynomial p_I^Q may not be multilinear, we cannot apply directly the techniques in [16]. Instead, our approach is to characterize the circuit bounds for a query Q via some other self-join-free CQ for which we have existing upper and lower bounds.

Recall that given a CQ Q, we define $[Q]$ to be the corresponding self-join-free CQ where every relational symbol is distinct. For example, for the Boolean CQ $Q() \leftarrow R(x,y) \wedge R(y,z) \wedge R(z,x)$, we have $[Q]() \leftarrow R_1(x,y) \wedge R_2(y,z) \wedge R_3(z,x)$. We first prove that constructing a circuit for $[Q]$ is at least as hard as constructing a circuit for Q.

Theorem 4.1. *Let $\mathbb{S}$ be any semiring and Q be a Boolean CQ. Then,* $\mathbb{S}(Q,N) \leq \mathbb{S}([Q],N)$.

Proof. Let I be an instance for Q such that every relation has size at most N. We construct an instance J for $[Q]$ as follows: for each tuple $R(\mathbf{a}) \in I$, we add a tuple $R_i(\mathbf{a})$ to J for each atom R_i in $[Q]$. It is easy to see that the size of every relation in J is still bounded by N. Now, let F' be the smallest circuit that computes the polynomial $p_J^{[Q]}$ over $\mathbb{S}$. By definition, we have that $|F'| \leq \mathbb{S}([Q],N)$. We will construct a circuit F for p_I^Q as follows: it is identical to F' but identifies the input variables that come from the same tuple in I (say $R_1(\mathbf{a}), R_2(\mathbf{a}), \dots$) to the same variable in I (the variable for the tuple $R(\mathbf{a})$). We argue that the constructed circuit F correctly computes p_I^Q over $\mathbb{S}$. Indeed, the circuit F computes the polynomial $p_J^{[Q]}$ under the constraint that are variables that correspond to the same tuple in I (say $R_1(\mathbf{a}), R_2(\mathbf{a}), \dots$) must be equal to the variable $x_{R(\mathbf{a})}$. But this will give us a circuit for the polynomial p_I^Q. □

Note that since Q might contain self-joins, the circuit F in the above proof is not necessarily multilinear and homogeneous. The converse of Theorem 4.1 does not hold in general, but we show that it holds under certain restrictions.

Theorem 4.2. *Let $\mathbb{S}$ be an idempotent semiring. Let Q be a Boolean CQ s.t. every endomorphism of Q is an automorphism. Then,* $\mathbb{S}(Q, |Q| \cdot N) \geq \mathbb{S}([Q],N)$ *where $|Q|$ is the size of Q.*

Proof. First, note that Q cannot have two atoms of the form $R(\mathbf{x}), R(\mathbf{x})$, since then we would have a homomorphism to a subset of Q. Thus, every atom of Q is uniquely identified by its relational symbol R and variable vector $\mathbf{x}$. We will thus use $R_{\mathbf{x}}$ to talk about the relational symbol of $[Q]$ that corresponds to the atom $R(\mathbf{x})$ in Q.

Suppose J is an instance for $[Q]$ where every relation has size at most N. Consider the *tagged instance* I of Q defined as follows [4, 8, 9]: For any tuple $R_{\mathbf{x}}(\mathbf{a}) \in J$, we add a tuple $R(\mathbf{a} \cdot \mathbf{x})$ to I where $\mathbf{a} \cdot \mathbf{x}$ is interpreted as the vector with entries $(\mathbf{a}_i, \mathbf{x}_i)$. Clearly, in I every relation has size at most $|Q| \cdot N$, where $|Q|$ is the number of atoms in Q. Let F be a circuit that computes p_I^Q over $\mathbb{S}$. Let G be the circuit that, whenever a wire in F connects to $R(\mathbf{a} \cdot \mathbf{x})$, it instead connects to $R_{\mathbf{x}}(\mathbf{a})$.

To complete the proof, it suffices to prove that G correctly computes $p_J^{[Q]}$ over $\mathbb{S}$. For this, take any annotation $\mu : J \to \mathbb{S}$ and define the annotation $\nu : I \to \mathbb{S}$ such that $\nu(R(\mathbf{a} \cdot \mathbf{x})) = \mu(R_{\mathbf{x}}(\mathbf{a}))$. The following properties hold:

- $p^G(\mu) = p^F(\nu)$: this holds directly from the construction of F' from F.
- $p^F(\nu) = p_I^Q(\nu)$: this holds because F computes p_I^Q.

We will next show that $p_I^Q(\nu) = p_J^{[Q]}(\mu)$; this implies that $p_J^{[Q]}(\mu) = p^G(\mu)$. Indeed, consider a monomial m in p_I^Q with input tuples $R^1(\mathbf{a}_1 \cdot \mathbf{x}_1), R^2(\mathbf{a}_2 \cdot \mathbf{x}_2), \ldots$. This monomial corresponds to an output tuple, hence there exists a valuation ψ from the variables of Q to the domain of J. Since every value of the domain is a pair of constant and a variable, ψ implicitly defines an endomorphism, i.e., a homomorphism from Q to itself, which we call the *tagging map* [9]. Because of our assumption of Q, this endomorphism must also be an automorphism. But this implies that the monomial m' with tuples $R^1_{\mathbf{x}_1}(\mathbf{a}_1), R^2_{\mathbf{x}_2}(\mathbf{a}_2), \ldots$ exists in $p_J^{[Q]}$. For the other direction, it is easy to see that if a monomial m' with tuples $R^1_{\mathbf{x}_1}(\mathbf{a}_1), R^2_{\mathbf{x}_2}(\mathbf{a}_2), \ldots$ exists in $p_J^{[Q]}$, then a monomial m with input tuples $R^1(\mathbf{a}_1 \cdot \mathbf{x}_1), R^2(\mathbf{a}_2 \cdot \mathbf{x}_2), \ldots$ exists in p_I^Q, since any valuation $\psi : \mathrm{vars}(Q) \to D$ can be extended to a valuation $\psi' : \mathrm{vars}(Q) \to \mathrm{vars}(Q) \times D$ by using the identity automorphism (every variable maps to itself). It is now easy to see that $\nu(m) = \mu(m')$.

We have thus shown that for every monomial in $p_J^{[Q]}$ there is one monomial in p_I^Q with the same value, and vice versa. Since $\mathbb{S}$ is idempotent, this implies that $p_I^Q(\nu) = p_J^{[Q]}(\mu)$. Thus, we have $p^G(\mu) = p_J^{[Q]}(\mu)$ for any annotation $\mu : J \to \mathbb{S}$. This means $p^G \equiv_{\mathbb{S}} p_J^{[Q]}$ and thus G correctly computes $p_J^{[Q]}$ over $\mathbb{S}$. □

Example 4.3. As an application of Theorem 4.2, consider the Boolean CQ $\Delta() \leftarrow R(x,y) \wedge R(y,z) \wedge R(z,x)$. This query has three endomorphisms:

$$h_1 : x \to x, y \to y, z \to z \qquad h_2 : x \to y, y \to z, z \to x \qquad h_3 : x \to z, y \to x, z \to y.$$

All three endomorphisms are also automorphisms, and hence for Δ its optimal circuit is exactly characterized by the optimal circuit of its self-join-free counterpart $[\Delta]() \leftarrow R_{xy}(x,y) \wedge R_{yz}(y,z) \wedge R_{zx}(z,x)$ for any idempotent semiring $\mathbb{S}$.

On the other hand, consider the Boolean CQ $Q_\square() \leftarrow R(x,y) \wedge S(y,z) \wedge R(x,w) \wedge S(w,z)$. This query has the endomorphism $h : x \to x, y \to y, w \to y, z \to z$, which is not an automorphism (since the four atoms are mapped in only two atoms of the query). Indeed, $[\mathrm{Q}_\square]$ has circuit size $\Theta(N^{3/2})$ for the tropical semiring, while as we will see $Q_\square$ admits a linear-size circuit.

Though powerful, Theorem 4.2 is only applicable for queries that are "minimal", in the sense that no endomorphism mapping to strictly smaller subset exists. To derive meaningful circuit lower bounds, we consider semirings where equivalency between queries is well-behaved.

5 CQs with self-joins: Circuit Bounds for $\mathbf{C}_{\mathrm{hom}}$

Kostylev, Reutter and Salamon [29] classified the class of semirings where query containment is equivalent to the existence of a homomorphism between CQs. Formally,

Definition 5.1. $\mathbf{C}_{\mathrm{hom}}$ is the class of semirings that satisfy the following axioms:

(1) ($\otimes$-idempotence) $x \otimes x = x$;
(2) ($\mathbf{1}$-annihilation) $\mathbf{1} \oplus x = \mathbf{1}$.

Examples of semirings in $\mathbf{C}_{\mathrm{hom}}$ include $\mathbb{B}$, $\mathrm{PosBool}[X]$ [19] and any distributive lattice [18]. Note that any semiring that satisfies $\mathbf{1}$-annihilation is also $\oplus$-idempotent, since $x \oplus x = \mathbf{1} \otimes x \oplus \mathbf{1} \otimes x = (\mathbf{1} \oplus \mathbf{1}) \otimes x = \mathbf{1} \otimes x = x$ for any $x \in \mathbb{S}$, where the third equality is due to $\mathbf{1}$-annihilation.

Observe that for any polynomial p over $\mathbb{S} \in \mathbf{C}_{\mathrm{hom}}$, we lose no information by viewing p as the set of its monomials, i.e., ignoring coefficients due to $\mathbf{1}$-annihilation, and further viewing each monomial as the set of its variables, i.e., dropping exponents due to $\otimes$-idempotence. In other words, we can identify any polynomial as a set of set of variables, and we write $p \subseteq q$ for two polynomials

p, q in the sense of set containment. Finally, for any polynomial p, we denote $p_{\min}$ as the set of monomials not containing any other monomial in p. Note that $p_{\min}$ is unique for every p.

Example 5.2. The polynomial $p = 5x^3 + 2x^2y + 3y^3$ is identified as the set $\{\{x\}, \{x, y\}, \{y\}\}$ and $p_{\min}$ is identified as the set $\{\{x\}, \{y\}\}$.

We write $Q_2 \to Q_1$ if there exists a homomorphism from Q_2 to Q_1. The results in [29] can be stated as the following.

Theorem 5.3 (Kostylev, Reutter & Salamon [29]). *A semiring $\mathbb{S}$ is in $\mathbf{C}_{\text{hom}}$ if and only if for all CQs Q_1 and Q_2, we have that $(p_I^{Q_1})_{\min} \subseteq (p_I^{Q_2})_{\min}$ for any instance I if and only if $Q_2 \to Q_1$.*

It is straightforward to derive the following using their results.

Theorem 5.4. *Fix any semiring $\mathbb{S} \in \mathbf{C}_{\text{hom}}$. Then, two CQs Q_1 and Q_2 are semantically equivalent over $\mathbb{S}$ if and only if they are homomorphically equivalent.*

Proof. If Q_1 and Q_2 are homomorphically equivalent, then $Q_1 \to Q_2$ and $Q_2 \to Q_1$, which implies that $(p_I^{Q_1})_{\min} = (p_I^{Q_2})_{\min}$ for any instance I. Observe that $p \equiv_{\mathbb{S}} p_{\min}$ for any p. Therefore, $p_I^{Q_1} \equiv_{\mathbb{S}} p_I^{Q_2}$ and thus Q_1 and Q_2 are semantically equivalent over $\mathbb{S}$.

If Q_1 and Q_2 are semantically equivalent over $\mathbb{S}$, we have $p_I^{Q_1} \equiv_{\mathbb{S}} p_I^{Q_2}$. By Theorem 5.3, it suffices to show $(p_I^{Q_1})_{\min} = (p_I^{Q_2})_{\min}$. Suppose for contradiction that $(p_I^{Q_1})_{\min} \neq (p_I^{Q_2})_{\min}$. Then, there exists a set m_1 in $(p_I^{Q_1})_{\min}$ that is not in $(p_I^{Q_2})_{\min}$. Setting every variable in m_1 to be $\mathbf{1}$ and every other variable to be $\mathbf{0}$, since $p_I^{Q_1} \equiv_{\mathbb{S}} p_I^{Q_2}$, we conclude that there exists a subset $m_2 \subsetneq m_1$ that is in $(p_I^{Q_2})_{\min}$. But then setting every variable in m_2 to be $\mathbf{1}$ and every other variable to be $\mathbf{0}$, we have $(p_I^{Q_1})_{\min}$ evaluates to $\mathbf{0}$ while $(p_I^{Q_2})_{\min}$ evaluates to $\mathbf{1}$, a contradiction. The proof is now complete. □

For any CQ Q, *a minimal equivalent query* Q' of Q over a semiring $\mathbb{S}$ is any CQ such that $Q' =_{\mathbb{S}} Q$, with no strictly smaller query of itself having this property. It is well-known that all minimal equivalent queries over $\mathbb{B}$ are isomorphic [10]. The following lemma generalizes this fact to any semiring in $\mathbf{C}_{\text{hom}}$.

Lemma 5.5. *Let Q be any CQ and $\mathbb{S} \in \mathbf{C}_{\text{hom}}$ be a semiring. Then, all minimal equivalent queries of Q over $\mathbb{S}$ are isomorphic.*

Proof. Let Q_1 and Q_2 be two minimal equivalent queries of Q over $\mathbb{S}$. By Theorem 5.4, there exists a homomorphism h from Q_1 to Q_2 and there exists a homomorphism g from Q_2 to Q_1.

Now, observe that the mapping $h \circ g$ is an endomorphism for Q_2. However, this endomorphism must also be an automorphism. Indeed, if not, then we have a query Q_2' with less atoms than Q_2 that by Theorem 5.4 is equivalent to Q, violating the minimality assumption. Symmetrically, $g \circ h$ is an automorphism for Q_1. But this can happen only if h, g are isomorphisms. □

CQ Core. The unique (up to isomorphism) minimal equivalent CQ of Q is called the *core* of Q, denoted Q_{core}. As in the Boolean case, the core can be constructed by removing one or more atoms from Q (in fact, the core is the same for any semiring in $\mathbf{C}_{\text{hom}}$).

Lemma 5.6. *Let Q be any CQ and let $\mathbb{S} \in \mathbf{C}_{\text{hom}}$. Then, there exists a minimal equivalent query $Q_{\min}$ to Q over $\mathbb{S}$ that can be obtained from Q by removing zero or more atoms.*

Proof. Let $Q_{\min}$ be any minimal equivalent query of Q over $\mathbb{S}$ [5]. By Theorem 5.4, there exists a homomorphism h from Q to $Q_{\min}$ and a homomorphism from $Q_{\min}$ to Q. Consider the composed

[5]There must exist such a $Q_{\min}$ by the following argument: Note that Q is equivalent to itself. Therefore, by ordering in terms of inclusion and the number of atoms, the set of equivalent queries is partially ordered. Now apply Zorn's lemma.

mapping $g \circ h$ from Q to Q. Clearly $g \circ h$ is a homomorphism from Q to itself, and thus $g \circ h(Q)$ is a subquery of Q. It suffices to show that $g \circ h(Q)$ is equivalent to Q over $\mathbb{S}$. This follows by Theorem 5.4 by noting that the identity map of $g \circ h(Q)$ is a homomorphism to Q. □

We are now ready to state the main result of this subsection.

Theorem 5.7 (Main Result for $\mathbf{C}_{\text{hom}}$). *Let Q be any Boolean CQ and let $\mathbb{S} \in \mathbf{C}_{\text{hom}}$. Then,*

$$\mathbb{S}(Q, N) = \Theta\left(\mathbb{S}([Q_{core}], N)\right).$$

Proof. Since Q_{core} is the minimal equivalent query to Q over $\mathbb{S} \in \mathbf{C}_{\text{hom}}$ – in particular Q_{core} is an equivalent query to Q over $\mathbb{S} \in \mathbf{C}_{\text{hom}}$ – we have $\mathbb{S}(Q, N) = \mathbb{S}(Q_{\text{core}}, N)$. By Theorem 4.1, we know $\mathbb{S}([Q_{\text{core}}], N) \geq \mathbb{S}(Q_{\text{core}}, N) = \mathbb{S}(Q, N)$ and thus $\mathbb{S}(Q, N)) = O(\mathbb{S}([Q_{\text{core}}], N))$.

We next claim that every endomorphism of Q_{core} is an automorphism. Indeed, if an endomorphism h exists that is not an automorphism, then the image $h(Q_{\text{core}})$ is a strictly smaller equivalent query to Q than Q_{core}, a contradiction. Therefore, Theorem 4.2 implies $\mathbb{S}(Q, |Q_{\text{core}}| \cdot N) = \mathbb{S}(Q_{\text{core}}, |Q_{\text{core}}| \cdot N) \geq \mathbb{S}([Q_{\text{core}}], N)$. Observe that $|Q_{\text{core}}|$ is a constant in terms of data complexity and the quantity $\mathbb{S}(Q, |Q_{\text{core}}| \cdot N)$ is an expression with N being the only variable – specifically, the dependence on N is polynomial. Thus, we have $\mathbb{S}(Q, N)) = \Theta(\mathbb{S}(Q, |Q_{\text{core}}| \cdot N)) = \Omega(\mathbb{S}([Q_{\text{core}}], N))$. We conclude that $\mathbb{S}(Q, N) = \Theta\left(\mathbb{S}([Q_{\text{core}}], N)\right)$. □

Example 5.8. Consider the query Q_1 in the running example of Figure 1. For this query, its core is the query $Q_2() \leftarrow R(x, y) \wedge S(y, z)$. Hence, the optimal circuit size for Q_1 over any semiring $\mathbb{S} \in \mathbf{C}_{\text{hom}}$ is asymptotically equal to that of $[Q_2]() \leftarrow R(x, y) \wedge S(y, z)$, which has circuit size $\Theta(N)$ since it is an acyclic query. Note that constructing the circuit for $[Q_1]$ would give a bound of $\Theta(N^{3/2})$, since $[Q_1]$ encodes the 4-cycle query.

Recall that for a self-join free CQ Q and any idempotent semiring, we can construct a semiring circuit of size $O(N^{\text{entw}(Q)})$. Consider now the situation where we start with a Boolean CQ with self-joins Q and start removing atoms to reach to Q_{core}; let $(Q =)Q_1, Q_2, \ldots, Q_k (= Q_{\text{core}})$ be the queries in this sequence. Applying Theorem 5.7 will then result in semiring circuits (for a semiring in $\mathbf{C}_{\text{hom}}$) of size $O(N^{\text{entw}([Q_1])}), O(N^{\text{entw}([Q_2])}), \ldots, O(N^{\text{entw}([Q_k])})$. We will show that in this case the entropic width must decrease, hence we obtain smaller and smaller circuits.

Theorem 5.9. *Let Q be a Boolean CQ and h be an endomorphism for Q. Then:*

$$\mathit{entw}([h(Q)]) \leq \mathit{entw}([Q]).$$

Proof. We first generalize the technique in the proof of Lemma 9 in [12] which constructs a tree decomposition of $h(Q)$ from that of Q, yet $h(Q)$ needs not be the core of Q as in [12]. W.l.o.g., we can assume that the variables in $h(Q)$ are mapped to themselves, i.e., $h(x) = x$.

Let $(\mathcal{T}, \chi)$ be a tree decomposition for Q. Consider the following structure $(\mathcal{T}', \chi')$ for $h(Q)$ where $\mathcal{T}' = \mathcal{T}$ and $\chi'(v) = h(\chi(v))$. We claim that $(\mathcal{T}', \chi')$ is a tree decomposition for $h(Q)$. Indeed, for any $e' \in \mathcal{E}(h(Q))$, there exists $e \in \mathcal{E}(Q)$ such that $h(e) = e'$, and thus e' appears in a bag of $(\mathcal{T}', \chi')$. Suppose two bags containing $v' \in \mathcal{V}(h(Q))$ are disconnected in $(\mathcal{T}', \chi')$ for the sake of contradiction. But by our assumption v' also is in $\mathcal{V}(Q)$, and thus is must be connected in $(\mathcal{T}, \chi)$. This proves our claim.

Let $\mathcal{H}$ and $\mathcal{H}'$ be the hypergraphs associated to Q and $h(Q)$, respectively. Recall that $\text{entw}(\mathcal{H}) := \max_{b \in \overline{\Gamma}^*_n \cap \text{ED}_{\mathcal{H}}} \min_{(\mathcal{T}, \chi) \in \text{TD}} \max_{v \in V(\mathcal{T})} b(\chi(v))$ where n is the number of variables in $\mathcal{H}$. Therefore, it suffices to show the following statement: let n and n' be the number of variables in Q and $h(Q)$ respectively, then for every $b' \in \overline{\Gamma}^*_{n'} \cap \text{ED}_{\mathcal{H}'}$, there exists $b \in \overline{\Gamma}^*_n \cap \text{ED}_{\mathcal{H}}$ such that $b'\text{-width}(\mathcal{H}') \leq b\text{-width}(\mathcal{H})$.

Let b' be any function in $\overline{\Gamma}^*_{n'} \cap \mathsf{ED}_{\mathcal{H}'}$, consider the function b as the entropy for the joint distribution over $\mathcal{V}(Q)$ where the random variables v for any $v \in \mathcal{V}(h(Q))$ are copies of the random variable v in b'. Clearly, b is entropic and $b \in \overline{\Gamma}^*_n \cap \mathsf{ED}_{\mathcal{H}}$. For any tree decomposition $(\mathcal{T}, \chi)$ for Q, observe that the b-width of Q for $(\mathcal{T}, \chi)$ is equal to the b'-width of $h(Q)$ for $(\mathcal{T}', \chi')$ constructed as above. Indeed, for any bag $\chi(t)$ and $\chi'(t)$ in $\mathcal{T}$ and $\mathcal{T}'$, we have $b(\chi(t)) = b'(\chi(t))$, by the definition of b. This implies that b'-width$(\mathcal{H}') \leq b$-width$(\mathcal{H})$ and thus finishes the proof. □

We remark that the existence of an endomorphism is crucial for Theorem 5.9, and just removing atoms from a query does not necessarily decrease the entropic width. Indeed, consider the Boolean CQs $Q() \leftarrow R(x,y) \wedge S(y,z) \wedge T(z,x) \wedge U(x,y,z)$ and $Q'() \leftarrow R(x,y) \wedge S(y,z) \wedge T(z,x)$. The number of atoms in Q' is strictly smaller than that in Q, yet we cannot conclude $\mathsf{entw}(Q') \leq \mathsf{entw}(Q)$ since there is no homomorphism from Q to Q'. In fact, $\mathsf{entw}(Q) = 1$ since it is acyclic, but $\mathsf{entw}(Q') = 1.5$.

6 CQs with self-joins: Circuit Bounds for $\mathbf{C}_=$

In this section, we study the behavior of CQs with self-joins for a different class of semirings, which we call $\mathbf{C}_=$. Given two polynomials p_1 and p_2 over the same set of variables, we write $p_1 \simeq p_2$ when the sets of monomials in p_1 and p_2 are identical (thus ignoring coefficients).

Definition 6.1. $\mathbf{C}_=$ is the class of semirings $\mathbb{S}$ where $p_1 \equiv_{\mathbb{S}} p_2$ implies $p_1 \simeq p_2$.

Any circuit F that computes a polynomial p over $\mathbb{S} \in \mathbf{C}_=$ must produce all monomials in p, i.e. $p^F \simeq p$. Notable examples of $\mathbf{C}_=$ are the tropical semiring $\mathbb{T}$, the counting semiring $\mathbb{C}$ and $\mathbb{B}[X]$ [20, 22, 29]. We also consider the *arithmetic semiring* where we overload the notation $\mathbb{Q} := (\mathbb{Q}_{\geq 0}, +, \cdot, 0, 1)$ whose underlying set is the set of non-negative rational numbers. The class $\mathbf{C}_=$ contains the class of semirings $\mathbf{C}_{\mathsf{bi}}$ defined in [29], as we show in Appendix A. However, we are unable to show the reverse direction of containment, nor can we find a counterexample. We thus leave the open problem whether $\mathbf{C}_=$ equals to $\mathbf{C}_{\mathsf{bi}}$.

The core of our lower bound construction for this section uses the following celebrated theorem, which says that computing all the partial derivatives of any polynomial collectively is asymptotically no harder than computing the polynomial itself [6]. We overload the notation $\mathbb{S}(p_1, \ldots, p_k)$ to mean the size of the optimal semiring circuit that computes all polynomials $p_1, p_2, \ldots, p_k$, with one output gate for every polynomial.

Theorem 6.2 (Baur & Strassen BaurS83). *For any polynomial p with variables $x_1, x_2, \ldots, x_n$, $\mathbb{C}(\partial_1(p), \partial_2(p), \ldots, \partial_n(p)) \leq 5 \cdot \mathbb{C}(p)$, where $\partial_i(p)$ is the partial derivative of p w.r.t. x_i, $1 \leq i \leq n$.*

We now state and prove the main result in this section. This result says that the circuit complexity of a CQ Q under any idempotent semiring in $\mathbf{C}_=$ is characterized by the circuit complexity of its self-join-free counterpart $[Q]$. This is even in the case when Q is not minimal in the Boolean sense.

Theorem 6.3 (Main Result for $\mathbf{C}_=$). *For a Boolean CQ Q and an idempotent semiring $\mathbb{S} \in \mathbf{C}_=$:*

$$\mathbb{S}(Q, N) = \Theta\left(\mathbb{S}([Q], N)\right).$$

Proof. By Theorem 4.1, it suffices to show that $\mathbb{S}([Q], N) = O(\mathbb{S}(Q, N))$. Let J be an instance for $[Q]$ where every relation has size at most N and consider the tagged instance I of Q as in the proof of Theorem 4.2. It is easy to see that in I every relation has size $O(N)$. Let F be a circuit that computes p_I^Q over $\mathbb{S}$ and let F' be the circuit that, whenever a wire in F connects to $R(\mathbf{a} \cdot \mathbf{x})$, it connects to $R_{\mathbf{x}}(\mathbf{a})$, as in the proof of Theorem 4.2. So far, it may not be true that F' computes the polynomial $p_J^{[Q]}$ over $\mathbb{S}$ correctly. Thus, we are going to construct another circuit F'' for to do so.

Before this, we will analyze the structure of p_I^Q and $p_J^{[Q]}$. To make the argument concrete, we will follow along an example. Consider the query $Q() \leftarrow R(u,v) \wedge R(u,w)$, $[Q]() \leftarrow R_{uv}(u,v) \wedge$

$R_{uw}(u, w)$ and $J = \{R_{uv}(a, b), R_{uw}(a, c)\}$. Then, the tagged instance is $I = \{R(au, bv), R(au, cw)\}$. The two polynomials are:

$$p_J^{[Q]} = x_{ab} \cdot x_{ac}$$
$$p_I^Q = 2x_{(au,bv)} \cdot x_{(au,cw)} + x_{(au,bv)}^2 + x_{(au,cw)}^2$$

Generally, each monomial m in p_I^Q falls into one of the two categories:

(1) Its term appears in $p_J^{[Q]}$ with coefficient 1 and its coefficient in p_I^Q equals to $|\mathsf{Aut}(Q)|$, the number of automorphisms of Q. For the running example, this would be the monomial $x_{(au,bv)} \cdot x_{(au,cw)}$.
(2) Its term does not appear in $p_J^{[Q]}$ and it corresponds to an endomorphism of Q that is not an automorphism. For our example, these would be the other two monomials.

Note that every monomial in p_I^Q must have the same degree. Therefore, any monomial that is of category (2) must have some variable of degree strictly greater than 1. In contrast, all monomials from category (1) must have all variables of degree 1. Moreover, by the construction of F and F', and also the property of $\mathbf{C}_=$, we know that $p^F \simeq p_I^Q$, i.e., the circuit F produces exactly the monomials in p_I^Q, possibly with different coefficients.

We now do apply the method introduced by Komerath, Randey and Rahul [28] that allows one to "extract" from the circuit only the monomials from category (1). For any input variable x annotating a tuple from relation $R_{\mathbf{x}}$ in F', we replace it with $w_{R_{\mathbf{x}}} \otimes x$ by introducing a $\otimes$-gate and a new variable $w_{R_{\mathbf{x}}}$. Call this new circuit $F^{(2)}$. Continuing our example, this would make the polynomial p_Q^I to compute instead the polynomial

$$2w_{R_{uv}} \cdot w_{R_{uw}} \cdot x_{(au,bv)} \cdot x_{(au,cw)} + w_{R_{uv}}^2 \cdot x_{(au,bv)}^2 + w_{R_{uw}}^2 \cdot x_{(au,cw)}^2$$

We now view the circuit $F^{(2)}$ as a circuit over $\mathbb{C}$ by interpreting $\oplus$-gates as +-gates and $\otimes$-gates as $\times$-gates, and apply Theorem 6.3 $|\mathcal{E}|$ many times to construct a circuit $F^{(3)}$ the polynomial $\frac{\partial^{|\mathcal{E}|}}{\partial w_{e_1} \ldots \partial w_{|\mathcal{E}|}} F^{(2)}$, where $\mathcal{E}$ is the set of hyperedges of the hypergraph corresponding to $[Q]$. Note that $F^{(3)}$ is a circuit over $\mathbb{C}$. Finally, we view $F^{(3)}$ as a circuit over $\mathbb{S}$ by interpreting +-gates as $\oplus$-gates and $\times$-gates as $\otimes$-gates, and set all w_i's, $1 \le i \le |\mathcal{E}|$, to be $\mathbf{0}$. Call this circuit $F^{(4)}$. After differentiating twice in the running example w.r.t. $w_{R_{uv}}, w_{R_{uw}}$, we would obtain a circuit that computes:

$$2 \cdot x_{(au,bv)} \cdot x_{(au,cw)} + 2w_{R_{uv}} \cdot x_{(au,bv)}^2 + 2w_{R_{uw}} \cdot x_{(au,cw)}^2$$

Setting finally $w_{R_{uv}}, w_{R_{uw}}$ to $\mathbf{0}$ would obtain a circuit that computes the polynomial $2 \cdot x_{(au,bv)} \cdot x_{(au,cw)}$, which is the desired quantity modulo the coefficient.

We now argue $F^{(4)}$ correctly computes $p_J^{[Q]}$. Indeed, all monomials from category (2) vanish since after $|\mathcal{E}|$ partial differentiations, at least one of the weight variables $w_{R_{\mathbf{x}}}$ will remain and therefore the entire monomial becomes $\mathbf{0}$. For all monomials from category (1), since every variable has degree 1 and comes from a different relation $R_{\mathbf{x}}$, their coefficient becomes a positive natural number. Since $\mathbb{S}$ is an idempotent semiring, we conclude that $F^{(4)}$ correctly computes $p_J^{[Q]}$. □

Since $\mathbb{T}$ and $\mathbb{B}[X]$ are both idempotent and lie in $\mathbf{C}_=$, we immediately obtain:

Corollary 6.4. *Let Q be any Boolean CQ. Then, $\mathbb{T}(Q, N) = \Theta(\mathbb{T}([Q], N))$ and $\mathbb{B}[X](Q, N) = \Theta(\mathbb{B}[X]([Q], N))$.*

The proof of Theorem 6.3 can be adapted to show the following result for $\mathbb{Q}$, which is in $\mathbf{C}_=$ but is not an idempotent semiring:

Theorem 6.5. *Let Q be any Boolean CQ. Then, $\mathbb{Q}(Q, N) = \Theta(\mathbb{Q}([Q], N))$.*

Proof. We perform the same construction of $F^{(4)}$ as in the proof of Theorem 6.3 and analyze the polynomial produced by $F^{(4)}$. All monomials from category (2) will still vanish, while every monomial from category (1) has the same coefficient $|\mathrm{Aut}(Q)|$. Therefore, we simply modify $F^{(4)}$ by adding a $\times$-gate having the output gate of $F^{(4)}$ and the scalar $\frac{1}{|\mathrm{Aut}(Q)|}$ as its input. □

7 CQs with self-joins: Circuit Bounds for $\mathbb{T}^+$

In this section, we consider the circuit complexity of CQ with self-joins over the *positive tropical semiring* $\mathbb{T}^+ := (\mathbb{R}^+ \cup \{+\infty\}, \min, +, +\infty, 0)$. $\mathbb{T}^+$ is idempotent and lies in neither $\mathbf{C}_{\mathrm{hom}}$ nor $\mathbf{C}_{=}$. $\mathbb{T}^+$ is an important semiring, because it captures finding the "minimum weight" valuation of a query.

Canonical Databases. We introduce here the notion of a *canonical database* [11, 18, 29], which is used to characterize query containment. A canonical database $[\![Q]\!]$ of a CQ Q is the instance for Q where every atom $R(\mathbf{x})$ has a tuple $\mathbf{x}$ annotated with $R^{[\![Q]\!]}(\mathbf{x}) = x_1 + x_2 + \cdots + x_n$ where $n \geq 0$ is the number of atoms in Q of the form $R(\mathbf{x})$ and x_i's are fresh variables.

Example 7.1. The canonical database for the Boolean $Q() \leftarrow R(u, v) \wedge R(u, w) \wedge S(v, w) \wedge S(v, w)$ is $R^{[\![Q]\!]}(u, v) = x_1$, $R^{[\![Q]\!]}(u, w) = x_2$ and $S^{[\![Q]\!]}(v, w) = x_3 + x_4$.

Given a Boolean CQ Q, we define its *canonical polynomial* p^Q as the sum-product polynomial $p^Q_{[\![Q]\!]}$ of Q over its canonical database $[\![Q]\!]$. For any self-join-free Q, this polynomial consists of a single term, where each variable in the term corresponds to a relation in Q.

Example 7.2. The canonical database for Q_1 in Figure 1a is $R^{[\![Q_1]\!]}(x, y) = x_1$, $R^{[\![Q_1]\!]}(x, w) = x_2$, $S^{[\![Q_1]\!]}(y, z) = z_1$, $S^{[\![Q_1]\!]}(w, z) = z_2$. The canonical polynomial for Q_1 is

$$p^{Q_1} := 2x_1x_2z_1z_2 + (x_1z_1)^2 + (x_2z_2)^2.$$

We say a monomial m in p^Q *sees an endomorphism of* Q, if when viewing m as a set of variables X_m, there exists an endomorphism h of Q such that X_m equals to the image of h. Every monomial in p^Q must see such an endomorphism of Q. The key idea is to find an equivalent expression for the canonical polynomial that is as simple as possible. We will do this via the following iterative procedure: for each monomial $m \in p^Q$, we check whether the polynomial p', which is the polynomial that we remove m from p^Q, is semantically equivalent to p^Q over $\mathbb{T}^+$; if so, set $p^Q = p'$. The next lemma shows that uniqueness of the output returned by this procedure.

Lemma 7.3. *There exists a sub-polynomial p^Q_{core} of p^Q such that the above procedure will always return p^Q_{core}, regardless of the order of picking monomials in p^Q.*

To prove Lemma 7.3, we need the following result.

Lemma 7.4 (Proposition 1.1 in [21]). *For any d-variate polynomial p over $\mathbb{T}^+$, the set defined as $\mathcal{D}(p) := \{(x, s) \in (\mathbb{R}^+)^{d+1} \mid x \in (\mathbb{R}^+)^d, s \in \mathbb{R}^+, s \leq p(x)\}$ is a full-dimensional closed convex polyhedron, i.e. of dimension $d + 1$.*

Even though [21] proves the case when the underlying set is $\mathbb{R}^+$, we can also adapt it to hold if we replace $\mathbb{R}^+$ by the positve rational numbers $\mathbb{Q}_{\geq 0}$. Indeed, all properties of $(\mathbb{R}^+)^{d+1}$ that are leveraged by the proof continue to hold in $\mathbb{Q}^{d+1}_{\geq 0}$. For example, the property that there exists a closed ball within any open ball also holds in $\mathbb{Q}^{d+1}_{\geq 0}$. However, we do not know whether a similar result also holds for the positive integers.

Proof of Lemma 7.3. We observe that the above procedure is nothing but picking the defining monomials for the facet of the polyhedron. Indeed, we can view the set $\mathcal{D}(p)$ in Lemma 7.4 as given

by linear inequalities, one for each monomial of the form $\sum a_i x_i \geq s$. Since for any full-dimensional polyhedron there exists a unique minimal set of defining inequalities [33], we conclude that p^Q_{core} is independent of the order of picking monomials in p^Q. □

Lemma 7.5. *Any polynomial $p \equiv_{\mathbb{T}^+} p^Q$ must contain p^Q_{core} as a sub-polynomial (i.e., all monomials of p^Q_{core} occur in p).*

Proof. By Lemma 7.4, we know that any polynomial p that is equivalent to p^Q over $\mathbb{T}^+$ must define the same full-dimensional closed convex polyhedron. Viewing them as a system of linear inequalities, this implies that both systems contain the same set of inequalities which defines the facets [33]. As in the proof of Lemma 7.3, p^Q_{core} represents the unique minimal set of defining inequalities, therefore p contains p^Q_{core} as a sub-polynomial. □

Based on the above, we can now define the notion of a *tropical core* for Q.

Definition 7.6 (Tropical Core). Given a Boolean CQ Q, its *tropical core* $\text{Tcore}(Q)$ is the set of CQs that are seen by the monomials in the polynomial p^Q_{core}.

Example 7.7. Consider the polynomial p^{Q_1} for our running example. Then, one can show that:

$$p^{Q_1}_{\text{core}} := (x_1 z_1)^2 + (x_2 z_2)^2.$$

Hence, the tropical core for Q_1 consists of the CQs $R(x,y) \wedge S(y,z)$ and $R(x,w) \wedge S(w,z)$, each one corresponding to an endomorphism of Q.

It is worth contrasting the tropical core to the case of $\mathbf{C}_{\text{hom}}$, where the core is a single CQ. Here, a tropical core may consist of multiple CQs, as the above example shows. We are now ready to characterize the circuit complexity of any CQ Q over $\mathbb{T}^+_{\mathbb{Q}}$.

Theorem 7.8 (Main Result for $\mathbb{T}^+$). *Let Q be any Boolean CQ. Then,*

$$\mathbb{T}^+(Q,N) = \Theta\left(\max_{Q^\star \in Tcore(Q)} \mathbb{T}^+([Q^\star],N)\right).$$

Since $\mathbb{T}^+([Q^\star],N) = \Theta(N^{\text{entw}}([Q^\star]))$, the circuit complexity of Q over $\mathbb{T}^+$ is characterized by the query $Q^\star$ in the tropical core with the highest entropic width.

We split the proof of Theorem 7.8 in two parts, the upper and lower bound.

Theorem 7.9 (Upper Bound). *Let Q be any Boolean CQ. Then,*

$$\mathbb{T}^+(Q,N) = O\left(\max_{Q^\star \in Tcore(Q)} \mathbb{T}^+([Q^\star],N)\right).$$

Proof. Consider an instance I with relation sizes at most N, and the polynomial p^Q_I for which we want to construct a circuit F. We will do this in two steps. First, we will construct a circuit $F_{Q^\star}$ for each query $Q^\star \in \text{Tcore}(Q)$. Second, we will compute the union of these circuits by taking the $\oplus$-sum of their output gates. In other words, we are essentially encoding in the circuit the union of a set of CQs.

We next describe the construction for $F_{Q^\star}$. Recall that each $Q^\star$ corresponds to an endomorphism $h^\star$ of Q. For every atom R in Q, let $N_R(h^\star)$ be the number of atoms from Q that $h^\star$ maps to R. As in the prior results, we construct an instance J for $[Q^\star]$ by adding for each tuple $R(\mathbf{a}) \in I$ a tuple $R_i(\mathbf{a})$ for each atom R_i in $[Q^\star]$. It is easy to see that the size of every relation in J is still bounded by N. We now construct a circuit $\hat{F}_{Q^\star}$ for $[Q^\star]$ that runs on J. To obtain $F_{Q^\star}$ from $\hat{F}_{Q^\star}$, we identify the input variable that comes from a fact $R_i(\mathbf{a})$ to the expression $(x_{R(\mathbf{a})})^{N_R(h^\star)}$; this is the variable

for the tuple $R(\mathbf{a}) \in I$ multiplied $N_R(h^\star)$ times. If $N_R(h^\star) = 0$, we can simply route this wire to $\mathbf{1}$. If $N_R(h^\star) > 0$, we compute the $\otimes$-product of $x_{R(\mathbf{a})}$ for as many times as prescribed. The size of the resulting circuit is still $O(\mathbb{T}^+([Q^\star], N))$.

We now argue that the resulting circuit correctly computes the desired polynomial p_I^Q. Given a tuple $t \in \bar{Q}(I)$, we define I_t to be the restriction of I on the image of the valuation that produces the tuple t. In other words, we get I_t by setting the annotation of any input tuple that does not contribute to t to $\mathbf{0}$. Kostylev et. al [29] showed that p_I^Q can be equivalently written as follows:

$$p_I^Q = \sum_{t \in \bar{Q}(I)} p_{I_t}^Q$$

Now let us look at the polynomial p^F produced by the circuit F. Every tuple $t \in \bar{Q}(I)$ can be traced back to a homomorphism h_t from Q to I_t. Now consider the query $Q^\star$. Since there is a homomorphism h from $Q^\star$ to Q (the identity homomorphism that maps each variable to itself), there exists a homomorphism $\hat{h}_t$ from $Q^\star$ to I_t. Hence, for each query $Q^\star$ in the tropical core, the polynomial p^F will contain a monomial $m^t_{Q^\star}$. In the other direction, every monomial in p^F corresponds to a homomorphism from some $Q^\star$ to I. But, since there is a homomorphism from Q to $Q^\star$, there is some tuple $t \in \bar{Q}(I)$ that corresponds to that monomial. Hence, p^F can be written as:

$$p^F = \sum_{t \in \bar{Q}(I)} \sum_{Q^\star \in \mathsf{Tcore}(Q)} m^t_{Q^\star}$$

To finish the proof, it suffices to show that for every tuple $t \in \bar{Q}(I)$, $p_{I_t}^Q \equiv_{\mathbb{T}} \sum_{Q^\star \in \mathsf{Tcore}(Q)} m^t_{Q^\star}$. Indeed, by the definition of the tropical core, it holds that for the canonical instance $[\![Q]\!]$ and its output tuple u,

$$p^Q_{[\![Q]\!]} \equiv_{\mathbb{T}} \sum_{Q^\star \in \mathsf{Tcore}(Q)} m^u_{Q^\star} \tag{3}$$

But since $[\![Q]\!]$ is canonical, there is a homomorphism from $[\![Q]\!]$ to I_t that can be viewed as a mapping ϕ of the variables in $p^Q_{[\![Q]\!]}$ to the variables in $p^Q_{I_t}$. For ϕ, we also have that $\phi(m^u_{Q^\star}) = m^t_{Q^\star}$. Hence, if we apply ϕ to Eq. (3) we obtain the desired result. □

Theorem 7.10 (Lower Bound). *Let Q be any Boolean CQ. Then, for any $Q^\star \in \mathsf{Tcore}(Q)$, we have that $\mathbb{T}(Q, N) = \Omega(\mathbb{T}^+([Q^\star], N)$.*

Proof. We follow the structure of the proof of Theorem 6.3.

Let J be an instance for $[Q^\star]$ where every relation has size at most N. Because $[Q^\star]$ is self-join-free, we will assume without loss of generality that the constants that correspond to distinct variables are distinct, in other words, no output tuple can take the same value for two different variables. Consider the tagged instance I of Q as in the proof of Theorem 4.2. It is easy to see that in I every relation has size $O(N)$. Let F be a circuit that computes p_I^Q over $\mathbb{T}^+$ and let F' be the circuit that, whenever a wire in F connects to $R(\mathbf{a} \cdot \mathbf{x})$, it connects to $R_{\mathbf{x}}(\mathbf{a})$, as in the proof of Theorem 4.2. So far, it may not be true that F' computes the desired polynomial correctly. Thus, we are going to construct another circuit F'' for to do so.

We will follow again a running example, this time with the query $Q() \leftarrow R(x, y) \wedge R(x, z) \wedge S(z, w) \wedge S(y, w)$, $[Q^\star]() \leftarrow R_{xy}(x, y) \wedge S_{yw}(y, w)$ and $J = \{R_{xy}(a, b), S_{yw}(b, c)\}$. Then, the tagged instance is $I = \{R(ax, by), S(by, cw)\}$. The two polynomials of interest are:

$$p_J^{[Q^\star]} = x_{ab} \cdot x_{bc}$$
$$p_I^Q = (x_{(ax,by)})^2 \cdot (x_{(by,cw)})^2$$

We now turn into the polynomial p^F. Note that, in contrast to our proof for $\mathbf{C}_=$, p^F may not be syntactically the same as p_I^Q. But we can now apply Lemma 7.5 to help us. Indeed, consider any tuple $t \in \bar{Q}(I)$. Consider the instance I_t as defined in the upper bound proof, and consider the polynomial $p_{I_t}^F$ produced by the circuit F if we set to $\mathbf{0}$ all the tuples that are not in I_t. For this polynomial, it must be that $p_{I_t}^F =_{\mathbb{T}^+} p_{I_t}^Q$. The instance I_t may not be isomorphic to the canonical instance $[\![Q]\!]$, but all variables in $Q^\star$ take distinct values in tuple t. Hence, since there is a homomorphism from $[\![Q]\!]$ to I_t and by Lemma 7.5, $p_{I_t}^F$ (and thus p^F) must contain a monomial that sees the endomorphism for $Q^\star$ for t. Moreover, every monomial in p^F that sees an endomorphism must correspond to some t, otherwise the circuit would fail to output the correct value.

We now apply the same differentiation method as in the proof of Theorem 6.3, but this time we differentiate on the variables that correspond to the atoms of $[Q^\star]$. Using the above reasoning, the only terms that will remain in the polynomial of the resulting circuit F'' will correspond to the ones that see the endomorphism for $Q^\star$ (any higher order terms that may exist will be zeroed out). In our running example, the polynomial after the differentiation will still be the same, i.e., we will have $p^{F''} = x_{(ax,by)}^2 \cdot x_{(by,cw)}^2$.

Our construction is not done yet, because the circuit F'' takes as an input powers of each variable, the power being the number of atoms of Q mapped to one of $Q^\star$ via the endomorphism. However, because every variable will appear with exactly the same power in every monomial, it is straighforward to factorize the circuit to another circuit F''' that computes the power of a variable before it is fed to the rest of the circuit. To obtain the desired circuit for $p_J^{[Q^\star]}$, we now simply have to replace the subcircuit that computes the power of a variable x, $(x)^n$, with one that takes only x as an input. □

8 Related Work

Circuit Size Bounds for CQs. The circuit complexity of query processing has been investigated in depth, and there has been a lot of research in the space of tractable circuits and knowledge compilation [2]. In this work, all circuits are tractable and we focus on a fine-grained study of the exact polynomial size depending on the query and the semiring. Closer to ours is the work of Fan, Koutris and Zhao [16] that proved circuit size bounds for self-join-free CQs over different semirings, which we leverage intensively in this paper. Wang and Yi [42] constructed non-monotone Boolean circuits for CQs under degree constraints by instantiating PANDA [26]. Their circuit sizes match the polymatroid bound up to polylogarithmic factors. Finally, recent work [32] explored the instance-optimality of circuits with fan-out 1 (which are formulas).

Enumeration of CQs with Self-Joins. Computing CQs with self-joins has been studied in depth. Of particular interest to this paper is the work of Carmeli-Kröll [8] and Carmeli-Segoufin [9], which introduced the technique of a tagged instance that we use. This line of work studies the enumeration complexity of CQs with self-joins.

Factorised Databases. It has been noted in [16] that d-representations and f-representations in Factorised Databases (FD) [5, 34, 35] can be seen as special cases of circuits and formulas for CQs where they are governed by exactly one tree decomposition. Our results in this paper can thus be regarded as results in FDs for CQs with self-joins in this sense.

Circuits for Polynomials over Semirings. The problem of finding circuit size bounds for a polynomial has been studied intensively in the literature. A line of work that is particularly related to our problem is done by Jukna [22, 23]. For example, the notion of "pure dynamic programming" in Jukna's language [23] translates to circuits over $\mathbb{T}$ in ours.

9 Conclusion

In this paper, we studied circuit size bounds for CQs with self-joins. We characterized the circuit size of CQ with self-joins by an appropriate self-join free CQ for some classes of semirings. Several open questions remain. First, the exact fine-grained circuit size bound for self-join-free CQs over $\mathbf{C}_=$ is still open [16]. Such a result will immediately transfer to tight circuit size bound for CQs with self-joins over $\mathbf{C}_=$ using our technique in Section 6. Another important question is whether we can close the gap between the upper and lower bound for self-join free CQs over $\mathbb{B}$. It is possible that one can prove this using probabilistic arguments that are not explored in this paper [1, 37, 38]. Also, it is interesting to see if our endomorphism point of view can be applied to other scenarios when studying CQs with self-joins [9, 17, 30, 36].

References

[1] Noga Alon and Ravi B. Boppana. 1987. The monotone circuit complexity of Boolean functions. *Comb.* 7, 1 (1987), 1–22.
[2] Antoine Amarilli and Florent Capelli. 2024. Tractable Circuits in Database Theory. *SIGMOD Rec.* 53, 2 (2024), 6–20.
[3] Albert Atserias, Anuj Dawar, and Phokion G. Kolaitis. 2006. On preservation under homomorphisms and unions of conjunctive queries. *J. ACM* 53, 2 (2006), 208–237.
[4] Guillaume Bagan, Arnaud Durand, and Etienne Grandjean. 2007. On Acyclic Conjunctive Queries and Constant Delay Enumeration. In *CSL (Lecture Notes in Computer Science, Vol. 4646)*. Springer, 208–222.
[5] Nurzhan Bakibayev, Tomás Kociský, Dan Olteanu, and Jakub Zavodny. 2013. Aggregation and Ordering in Factorised Databases. *Proc. VLDB Endow.* 6, 14 (2013), 1990–2001.
[6] Walter Baur and Volker Strassen. 1983. The Complexity of Partial Derivatives. *Theor. Comput. Sci.* 22 (1983), 317–330.
[7] Karl Bringmann and Egor Gorbachev. 2024. A Fine-grained Classification of Subquadratic Patterns for Subgraph Listing and Friends. *CoRR* abs/2404.04369 (2024).
[8] Nofar Carmeli and Markus Kröll. 2021. On the Enumeration Complexity of Unions of Conjunctive Queries. *ACM Trans. Database Syst.* 46, 2 (2021), 5:1–5:41.
[9] Nofar Carmeli and Luc Segoufin. 2023. Conjunctive Queries With Self-Joins, Towards a Fine-Grained Enumeration Complexity Analysis. In *PODS*. ACM, 277–289.
[10] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *STOC*. ACM, 77–90.
[11] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *STOC*. ACM, 77–90.
[12] Hubie Chen, Georg Gottlob, Matthias Lanzinger, and Reinhard Pichler. 2020. Semantic Width and the Fixed-Parameter Tractability of Constraint Satisfaction Problems. In *IJCAI*. ijcai.org, 1726–1733.
[13] Sara Cohen, Werner Nutt, and Yehoshua Sagiv. 2007. Deciding equivalences among conjunctive aggregate queries. *J. ACM* 54, 2 (2007), 5.
[14] Ronald Fagin and Phokion G. Kolaitis. 2012. Local transformations and conjunctive-query equivalence. In *PODS*. ACM, 179–190.
[15] Austen Z. Fan, Paraschos Koutris, and Hangdong Zhao. 2023. The Fine-Grained Complexity of Boolean Conjunctive Queries and Sum-Product Problems. In *ICALP (LIPIcs, Vol. 261)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 127:1–127:20.
[16] Austen Z Fan, Paraschos Koutris, and Hangdong Zhao. 2024. Tight Bounds of Circuits for Sum-Product Queries. *Proceedings of the ACM on Management of Data* 2, 2 (2024), 1–20.
[17] Cibele Freire, Wolfgang Gatterbauer, Neil Immerman, and Alexandra Meliou. 2020. New Results for the Complexity of Resilience for Binary Conjunctive Queries with Self-Joins. In *PODS*. ACM, 271–284.
[18] Todd J. Green. 2011. Containment of Conjunctive Queries on Annotated Relations. *Theory Comput. Syst.* 49, 2 (2011), 429–459.
[19] Todd J. Green, Gregory Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *PODS*. ACM, 31–40.
[20] Mark Jerrum and Marc Snir. 1982. Some Exact Complexity Results for Straight-Line Computations over Semirings. *J. ACM* 29, 3 (1982), 874–897.
[21] Michael Joswig. 2021. *Essentials of tropical combinatorics*. Vol. 219. American Mathematical Society.
[22] Stasys Jukna. 2015. Lower Bounds for Tropical Circuits and Dynamic Programs. *Theory Comput. Syst.* 57, 1 (2015), 160–194.
[23] Stasys Jukna. 2016. Tropical Complexity, Sidon Sets, and Dynamic Programming. *SIAM J. Discret. Math.* 30, 4 (2016), 2064–2085.

[24] Grigoris Karvounarakis and Todd J. Green. 2012. Semiring-annotated data: queries and provenance? *SIGMOD Rec.* 41, 3 (2012), 5–14.
[25] Mahmoud Abo Khamis, Hung Q. Ngo, and Atri Rudra. 2016. FAQ: Questions Asked Frequently. In *PODS*. ACM, 13–28.
[26] Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. 2017. What Do Shannon-type Inequalities, Submodular Width, and Disjunctive Datalog Have to Do with One Another?. In *PODS*. ACM, 429–444.
[27] Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. 2023. What do Shannon-type Inequalities, Submodular Width, and Disjunctive Datalog have to do with one another? arXiv:1612.02503
[28] Balagopal Komarath, Anurag Pandey, and Chengot Sankaramenon Rahul. 2022. Monotone Arithmetic Complexity of Graph Homomorphism Polynomials. In *ICALP (LIPIcs, Vol. 229)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 83:1–83:20.
[29] Egor V. Kostylev, Juan L. Reutter, and András Z. Salamon. 2012. Classification of annotation semirings over query containment. In *PODS*. ACM, 237–248.
[30] Paraschos Koutris, Xiating Ouyang, and Jef Wijsen. 2024. Consistent Query Answering for Primary Keys on Rooted Tree Queries. *Proc. ACM Manag. Data* 2, 2 (2024), 76.
[31] Paraschos Koutris and Jef Wijsen. 2021. Consistent Query Answering for Primary Keys in Datalog. *Theory Comput. Syst.* 65, 1 (2021), 122–178.
[32] Neha Makhija and Wolfgang Gatterbauer. 2024. Minimally Factorizing the Provenance of Self-join Free Conjunctive Queries. *Proc. ACM Manag. Data* 2, 2 (2024), 104.
[33] Peter McMullen. 2005. Convex Polytopes, by Branko Grunbaum, second edition (first edition (1967) written with the cooperation of V. L. Klee, M. Perles and G. C. Shephard. *Comb. Probab. Comput.* 14, 4 (2005), 623–626.
[34] Dan Olteanu and Maximilian Schleich. 2016. Factorized Databases. *SIGMOD Rec.* 45, 2 (2016), 5–16.
[35] Dan Olteanu and Jakub Závodný. 2015. Size Bounds for Factorised Representations of Query Results. *ACM Trans. Database Syst.* 40, 1 (2015), 2:1–2:44.
[36] Anantha Padmanabha, Luc Segoufin, and Cristina Sirangelo. 2024. A Dichotomy in the Complexity of Consistent Query Answering for Two Atom Queries With Self-Join. *Proc. ACM Manag. Data* 2, 2 (2024), 74.
[37] Ran Raz and Avi Wigderson. 1992. Monotone Circuits for Matching Require Linear Depth. *J. ACM* 39, 3 (1992), 736–744.
[38] Alexander Razborov. 1985. Lower bounds on the monotone complexity of some Boolean function. In *Soviet Math. Dokl.*, Vol. 31. 354–357.
[39] Neil Robertson and Paul D. Seymour. 1984. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B* 36, 1 (1984), 49–64.
[40] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic Circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.* 5, 3-4 (2010), 207–388.
[41] Balder ten Cate, Victor Dalmau, Phokion G. Kolaitis, and Wei-Lin Wu. 2024. When Do Homomorphism Counts Help in Query Algorithms?. In *ICDT (LIPIcs, Vol. 290)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:20.
[42] Yilei Wang and Ke Yi. 2022. Query Evaluation by Circuits. In *PODS*. ACM, 67–78.

A Connections between Semiring Classes

In this part, we show the connection between two classes of semirings, $\mathbf{C}_{\mathsf{bi}}$ and $\mathbf{C}_{=}$.

Definition A.1 ([29]). The class $\mathbf{C}_{\mathsf{bi}}$ contains all semirings $\mathbb{S}$ for which for any polynomial p and any monomial $x_1 \otimes x_2 \otimes \cdots \otimes x_n$, the inequality $x_1 \otimes x_2 \otimes \cdots \otimes x_n \preceq_{\mathbb{S}} p$ implies that p contains the monomial $x_1 \otimes x_2 \otimes \cdots \otimes x_n$.

Proposition A.2. $\mathbf{C}_{\mathsf{bi}} \subseteq \mathbf{C}_{=}$.

Proof. Note that all semirings in $\mathbf{C}_{\mathsf{bi}}$ are naturally ordered [29]. We show that if for any polynomial p and any monomial $x_1 \otimes x_2 \otimes \cdots \otimes x_n \preceq_{\mathbb{S}} p$, we have $x_1 \otimes x_2 \otimes \cdots \otimes x_n \in p$, then $p_1 \equiv_{\mathbb{S}} p_2$ implies $p_1 \simeq p_2$. Indeed, for any monomial $x_1 \otimes x_2 \otimes \cdots \otimes x_n \in p_1$, we have $x_1 \otimes x_2 \otimes \cdots \otimes x_n \preceq^{\mathsf{nat}}_{\mathbb{S}} p_1$ by the definition of $\preceq^{\mathsf{nat}}_{\mathbb{S}}$ and therefore $x_1 \otimes x_2 \otimes \cdots \otimes x_n \preceq_{\mathbb{S}} p_1 \equiv_{\mathbb{S}} p_2$ since $\preceq^{\mathsf{nat}}_{\mathbb{S}}$ is a subrelation of $\preceq_{\mathbb{S}}$ for any naturally ordered semiring. Therefore $x_1 \otimes x_2 \otimes \cdots \otimes x_n \in p_2$ and thus we conclude $p_1 \simeq p_2$ since the monomial $x_1 \otimes x_2 \otimes \cdots \otimes x_n$ is arbitrary. □

We do not know whether the reverse inclusion is also true and leave this as an open question. An affirmative answer would be interesting and quite surprising, since that allows one to characterize $\mathbf{C}_{\mathsf{bi}}$ without ever mentioning the partial relation.

B Circuit Bounds with Degree Constraints

In the main body of the paper, we only presented upper and lower circuit bounds when the only constraint is an upper bound N on the size of each relation. Here, we will show how to extend all of our upper and lower bounds to the case where we have more general constraints, called degree constraints.

We define the notion of *degree constraints* as in [27]. Let DC be a set of triples $(X, Y, N_{Y|X})$ for some $X \subset Y \subseteq [n]$ and $N_{Y|X} \in \mathbb{N}$. An instance I satisfies the degree constraints DC if for every relation R_e in I with $X \subseteq Y \subseteq e$ and every tuple t_X we have $|\pi_Y(R_e \ltimes t_X)| \leq N_{Y|X}$. A constraint of the form $(\emptyset, e, N_e)$ is simply a cardinality constraint. The degree constraints on an instance can be translated as constraints on entropic functions as follows:

$$\mathsf{HDC} := \left\{ h : 2^{[n]} \to \mathbb{R}_+ \mid \bigwedge_{(X,Y,N_{Y|X}) \in \mathsf{DC}} h(Y|X) \leq \log N_{Y|X} \right\}$$

where $h(Y|X) := h(Y) - h(X)$. For a given set of degree constraints HDC, we also define the "scaled-up" degree constraints $\mathsf{HDC} \times k$ as the same set of constraints but with all the degree bounds multiplied by k.

In analogy to the entropic and submodular width, we can define their degree-aware versions, called degree-aware entropic and degree-aware submodular width respectively:

$$\text{da-entw}(\mathcal{H}, \mathsf{HDC}) := \max_{h \in \overline{\Gamma}^*_n \cap \mathsf{HDC}} \min_{(\mathcal{T},\chi) \in \mathsf{TD}} \max_{v \in V(\mathcal{T})} h(\chi(v))$$

$$\text{da-subw}(\mathcal{H}, \mathsf{HDC}) := \max_{h \in \Gamma_n \cap \mathsf{HDC}} \min_{(\mathcal{T},\chi) \in \mathsf{TD}} \max_{v \in V(\mathcal{T})} h(\chi(v))$$

We can now generalize the theorems from [16] in terms of degree constraints.

Theorem B.1 ([16]). *For any $\epsilon > 0$ and any self-join-free CQ Q, there exists an instance I and $k > 0$ that satisfies the constraints* $\mathsf{HDC} \times k$ *such that any circuit F that computes the polynomial p^Q_I over a semiring in $\{\mathbb{B}_{\mathsf{lin}}, \mathbb{T}, \mathbb{C}\}$ has size*

$$\log |F| \geq (1 - \epsilon) \cdot \textit{da-entw}(Q, \mathsf{HDC} \times k).$$

Theorem B.2 ([16]). *Fix a self-join-free CQ Q and a set of degree constraints* DC. *Let I be any instance that satisfies* DC. *Then, there exists a multilinear and homogeneous circuit F of size $O(2^w)$ that produces the polynomial p_I^Q over any idempotent semiring, where $w = da\text{-}entw(Q, \mathsf{HDC})$.*

If DC is a set of degree constraints, we denote $\mathbb{S}(Q, \mathrm{DC})$ to be the optimal size of a circuit that computes the polynomial p_I^Q for a Boolean CQ Q over $\mathbb{S}$ over all instances I that satisfy the degree constraints. We now present the analogues of the results in the main body of the paper.

Theorem B.3. *Let $\mathbb{S}$ be any semiring and Q be a Boolean CQ. Then, for any degree constraints* DC, $\mathbb{S}(Q, \mathrm{DC}) \leq \mathbb{S}([Q], \mathrm{DC})$.

Proof. Observe that the proof of Theorem 4.1 preserves all degree constraints, since it simply creates copies of each relation. □

Theorem B.4. *Let $\mathbb{S}$ be any idempotent semiring, and Q be a Boolean CQ s.t. every endomorphism is an automorphism. Then, $\mathbb{S}(Q, \mathrm{DC} \times k) \geq \mathbb{S}([Q], \mathrm{DC})$, where k is the number of atoms in Q.*

Proof. The proof of Theorem 4.2 preserves the degree constraints if we scale them by a factor of k. □

Analogous results can be shown for all of our main theorems, since the tagging construction maintains the degree-aware constraints if we scale them by a constant factor that depends only on the query.

Received December 2024; revised February 2025; accepted March 2025