Generic Expression Hardness Results for Primitive Positive Formula Comparison

Simone Bova
Dept. of Mathematics
Vanderbilt University
Nashville, TN, USA
simone.bova@vanderbilt.edu

Hubie Chen
Dept. de Tecnologies
Universitat Pompeu Fabra
Barcelona, Spain
hubie.chen@upf.edu

Matthew Valeriote
Dept. of Mathematics & Statistics
McMaster University
Hamilton, Canada
matt@math.mcmaster.ca

Abstract

We study the expression complexity of three basic problems involving the comparison of primitive positive formulas. We give two generic hardness results for the studied problems, and discuss evidence that they are optimal.

1 Introduction

A *primitive positive (pp)* formula is a first-order formula defined from atomic formulas and equality of variables using conjunction and existential quantification. The class of primitive positive formulas includes, and is essentially equivalent to, the class of *conjunctive queries*, which is well-established in relational database theory as a pertinent and useful class of queries, and which has been studied complexity-theoretically from a number of perspectives (see for example [18, 16, 1]). In this paper, we study the complexity of the following fundamental problems, each of which involves the comparison of two pp-formulas ϕ, ϕ' having the same free variables, over a relational structure.

- Equivalence: are the formulas ϕ, ϕ' equivalent–that is, do they have the same satisfying assignments–over the structure?
- Containment: are the satisfying assignments of ϕ contained in those of ϕ', over the structure?
- Isomorphism: does there exist a permutation of the free variables for one of the formulas under which the two formulas are equivalent?

We study the complexity of these computational problems with respect to various fixed structures. That is, we parameterize each of these problems with respect to the structure to obtain a family of problems, containing one member for each structure, and study the resulting families of problems. To employ the terminology of Vardi [20], we study the *expression complexity* of the presented comparison tasks. The suggestion here is that various relational structures–which may represent databases or knowledge bases, according to use–may possess structural characteristics that affect the complexity of the resulting problems, and our interest is in understanding this interplay. The present work focuses on relational structures that are finite (that is, have finite universe), and we assume that the structures under discussion are finite.

Our study utilizes universal-algebraic tools that are currently being used to study computational problems related to primitive positive formulas, such as the *constraint satisfaction problem (CSP)*, which can be viewed as the problem of deciding if a primitive positive formula is satisfiable over a given structure. It is known that, relative to a structure, the set of relations that are definable by a primitive positive formula forms a robust algebraic object known as a *relational clone*; a Galois correspondence associates, in a bijective manner, each such relational clone with a *clone*, a set of operations with certain closure properties. This correspondence provides a way to pass from a relational structure $\mathbf{B}$ to an algebra $\mathbb{A}_{\mathbf{B}}$ whose set of operations is the mentioned clone, in such a way that two structures having the same algebra have the same complexity (for each of the mentioned problems). In a companion paper [6] by the present authors, we developed this correspondence and presented some basic complexity results for the problems at hand, including a classification of the complexity of the problems on all two-element structures.

In this paper, we present two general expression hardness results on the problems of interest. In particular, each of our two main results provides a sufficient condition on a structure so that the problems are hard for certain complexity classes. Furthermore, we give evidence that our results are optimal, in that the conditions that they involve in fact

describe dichotomies in the complexity of the studied problems. We now turn to describe each of our hardness results in greater detail.

Our first hardness result yields that for any structure $\mathbf{B}$ whose associated algebra $\mathbb{A}_\mathbf{B}$ gives rise to a variety $\mathcal{V}(\mathbb{A}_\mathbf{B})$ that *admits the unary type*, the equivalence and containment problems are Π_2^p-complete. Note that this is the maximal complexity possible for these problems, as the problems are contained in the class Π_2^p. The condition of admitting the unary type originates from tame congruence theory, a theory developed to understand the structure of finite algebras. We observe that this result implies a dichotomy in the complexity of the studied problems under the *G-set conjecture* for the CSP, a conjecture that predicts exactly where the tractability/intractability dichotomy lies for the CSP. In particular, under the G-set conjecture, the structures not obeying the described condition have equivalence and containment problems in coNP. The resolution of the G-set conjecture, on which there has been focused and steady progress over the past decade [10, 14, 11, 2], would thus, in combination with our hardness result, yield a coNP/Π_2^p-complete dichotomy for the equivalence and containment problems. For the isomorphism problem, we also demonstrate that the G-set conjecture would yield a dichotomy between two modes of complexity behavior that cannot coincide, unless the polynomial hierarchy collapses. In fact, this hardness result already unconditionally implies dichotomies for our problems for all classes of structures where the G-set conjecture has already been established, including the class of three-element structures [10], the class of conservative structures [7], and the class of undirected graphs [8].

One formulation of the G-set conjecture is that, for a structure $\mathbf{B}$ whose associated algebra $\mathbb{A}_\mathbf{B}$ is idempotent, the absence of the unary type in the variety generated by $\mathbb{A}_\mathbf{B}$ implies that $\mathsf{CSP}(\mathbf{B})$ is polynomial-time tractable. The presence of the unary type is a known sufficient condition for intractability in the idempotent case [9, 11], and this conjecture predicts exactly where the tractability/intractability dichotomy lies for the CSP. It should be noted, however, that the boundary that is suggested by our hardness result for the equivalence, containment, and isomorphism problems is *not* the same as the boundary suggested by the G-set conjecture for the CSP. The G-set conjecture, which is typically phrased on idempotent algebras, yields a prediction on the CSP complexity of all structures via a theorem [9] showing that each structure $\mathbf{B}$ has the same CSP complexity as a structure $\mathbf{B}'$ whose associated algebra is idempotent. The mapping from $\mathbf{B}$ to $\mathbf{B}'$ does not preserve the complexity of our problems, and indeed, there are examples [6] of two-element structures $\mathbf{B}$ such that our hardness result applies to $\mathbf{B}$–the equivalence and containment problems on $\mathbf{B}$ are Π_2^p-complete–but $\mathbf{B}'$ does not admit the unary type and indeed has a polynomial-time tractable CSP [6]. Our new result requires establishing a deeper understanding of the identified algebras' structure, some of which are CSP tractable, in order to obtain hardness.

Our second hardness result, which concerns structures $\mathbf{B}$ whose associated algebra $\mathbb{A}_\mathbf{B}$ is idempotent, implies that for any such structure, if the variety $\mathcal{V}(\mathbb{A}_\mathbf{B})$ is not congruence modular, then the equivalence and containment problems are coNP-hard. Note that structures having all constants are well-known to have idempotent algebras, and so this result covers such structures.[1] The question of whether or not this second hardness result can be extended to all structures is left as a tantalizing open question.

Previous work identified one most general condition for the tractability of the equivalence and containment problems: if the algebra *has few subpowers*–a combinatorial condition [3, 14] involving the number of subalgebras of powers of an algebra–then these problems are polynomial-time tractable. This second hardness result appears to perfectly complement this tractability result: there are no known examples of algebras $\mathbb{A}_\mathbf{B}$ (of structures $\mathbf{B}$ having finitely many relations) that are not covered by one of these results, and in fact the *Edinburgh conjecture* predicts that none exist, stating that every such algebra $\mathbb{A}_\mathbf{B}$ that generates a congruence modular variety also has few subpowers. Concerning this conjecture, it should be pointed out, on an optimistic note, that the resolution of the *Zadori conjecture*, a closely related conjecture of which the Edinburgh conjecture is a generalization, was recently announced by L. Barto. We also point out that this conjecture (as with the Zadori conjecture) is purely algebraic, making no references to notions of computation. This second hardness result similarly suggests a dichotomy for the isomorphism problem.

All put together, the picture that emerges from this work is a trichotomy in the complexity of the studied problems. In the case of equivalence and containment, one appears to have Π_2^p-completeness for an algebra with a variety admitting the unary type; coNP-completeness for an algebra with a non-congruence modular variety omitting the unary type, and polynomial-time tractability otherwise, with a picture for isomorphism that is similar but shifted slightly upwards. We certainly look forward to future work on these problems and the related challenging conjectures.

2 Preliminaries

Here, a *signature* is a set of relation symbols, each having an associated arity; we assume that all signatures are of finite size. A *relational structure* over a signature σ consists of a universe B and, for each relation symbol $R \in \sigma$, a relation $R^\mathbf{B} \subseteq B^k$ where k is the arity of R. We assume that

[1] By constants, here we mean singleton unary relations.

all relational structures under discussion have universes of finite size. A *primitive positive formula (pp-formula)* on σ is a first-order formula formed using equalities on variables $(x = x')$, atomic formulas $R(x_1, \ldots, x_k)$ over σ, conjunction $(\wedge)$, and existential quantification $(\exists)$.

We now define the problems that will be studied. For the isomorphism problem, we will make use of the following notion. A *sectioning* for a formula with free variables X is a pair of mappings $(s : X \to S, t : X \to T)$ such that the pair mapping $(s, t) : X \to S \times T$ is a bijection. Let ϕ, ϕ' be formulas with free variables X_ϕ, $X_{\phi'}$ and sectionings (s, t), (s', t'), respectively. We say that a bijection $\pi : X_{\phi'} \to X_\phi$ *respects* the sectionings if for all $x, x_1, x_2 \in X_{\phi'}$, (1) $s'(x_1) = s'(x_2)$ if and only if $s(\pi(x_1)) = s(\pi(x_2))$, and (2) $t'(x) = t(\pi(x))$. One can think of $s(x)$ as the section of a variable x, and of $t(x)$ as its type; the requirement on (s, t) requires that each section has exactly one variable of each type, and the given notion of respecting requires that sections are mapped to sections in a type-preserving way.

Definition 2.1 We define the following computational problems. For each of the first two, an instance consists of a relational structure $\mathbf{B}$ and a pair (ϕ, ϕ') of pp-formulas over the signature of $\mathbf{B}$ having the same set of free variables X; for the third problem (PPISO), an instance consists of these objects and, in addition, sectionings (s, t) and (s', t') for the formulas.

- PPEQ: decide if ϕ and ϕ' are equivalent, that is, whether for all $f : X \to B$, it holds that $\mathbf{B}, f \models \phi$ iff $\mathbf{B}, f \models \phi'$.
- PPCON: decide if ϕ is contained in ϕ', that is, whether for all $f : X \to B$, it holds that $\mathbf{B}, f \models \phi$ implies $\mathbf{B}, f \models \phi'$.
- PPISO: decide if ϕ and ϕ' are isomorphic, relative to the given sectionings, that is, whether there exists a bijection $\pi : X \to X$ respecting the sectionings such that for all $f : X \to B$, it holds that $\mathbf{B}, f \models \phi$ if and only if $\mathbf{B}, f \circ \pi \models \phi'$.

For every relational structure $\mathbf{B}$, we define PPEQ($\mathbf{B}$) to be the problem PPEQ where the structure is fixed to be $\mathbf{B}$; hence, an instance of PPEQ($\mathbf{B}$) is just a pair (ϕ, ϕ') of pp-formulas. We define the problems PPCON($\mathbf{B}$) and PPISO($\mathbf{B}$) similarly. □

It is straightforward to verify that the PPEQ and PPCON problems are contained in Π_2^p, and that the PPISO problem is contained in Σ_3^p.

We use the described formulation of the PPISO problem as it is robust with respect to changes in representation, as shown, for example, by the following proposition. We use BOOL-PPISO to denote the restriction of the PPISO to instances where the structure $\mathbf{B}$ is boolean (two-element). Throughout the paper, the notion of reduction used is logspace many-one reducibility.

Proposition 2.2 *The problem* PPISO *reduces to the problem* BOOL-PPISO.

Proof. See Appendix A. □

In this paper, we will overload problems such as PPISO and use a problem to denote the set of all problems that reduce to it. This will allow us to talk about, for instance, PPISO-completeness.

Proposition 2.3 *For each structure* $\mathbf{B}$*, the problem* PPCON($\mathbf{B}$) *reduces to the problem* PPEQ($\mathbf{B}$).

Proof. The reduction, given an instance (ϕ, ϕ') of PPCON($\mathbf{B}$), outputs the instance $(\phi, \phi \wedge \phi')$ of PPEQ($\mathbf{B}$). □

We now review the relevant algebraic concepts to be used. An *algebra* is a pair $\mathbb{A} = (A, F)$ such that A is a nonempty set, called the *domain* or *universe* of the algebra, and F is a set of finitary operations on A. Let $\mathbb{A} = (A, F)$ be an algebra; a *term operation* of $\mathbb{A}$ is a finitary operation obtained by composition of (1) operations in F and (2) projections on A, and a *polynomial operation* is a finitary operation obtained by composition of (1) operations in F, (2) projections on A and (3) constants from A. An operation $f(x_1, \ldots, x_n)$ on A is said to be *idempotent*if the equality $f(a, a, \ldots, a) = a$ holds for all $a \in A$. An algebra $\mathbb{A}$ is *idempotent* if all of its term operations are.

Let B be a nonempty set, let f be an n-ary operation on B, and let R be a k-ary relation on B. We say that f *preserves* R (or f is a *polymorphism* of R), if for every length n sequence of tuples $t_1, \ldots, t_n \in R$, denoting the tuple t_i by $(t_{i,1}, \ldots, t_{i,k})$, it holds that the tuple $f(t_1, \ldots, t_n) = (f(t_{1,1}, \ldots, t_{n,1}), \ldots, f(t_{1,k}, \ldots, t_{n,k}))$ is in R. We extend this terminology to relational structures: an operation f is a polymorphism of a relational structure $\mathbf{B}$ if f is a polymorphism of every relation of $\mathbf{B}$. We use Pol($\mathbf{B}$) to denote the set of all polymorphisms of a relational structure $\mathbf{B}$, and use $\mathbb{A}_\mathbf{B}$ to denote the algebra $(B, \mathsf{Pol}(\mathbf{B}))$. We remark that it is well-known and straightforward to verify that a relational structure $\mathbf{B}$ having all constants (singleton unary relations) has an idempotent algebra $\mathbb{A}_\mathbf{B}$. Dually, for an operation f, we use Inv(f) to denote the set of all relations that are preserved by f, and for a set of operations F, we define Inv(F) as $\bigcap_{f \in F} \mathsf{Inv}(f)$. We will make use of the following result connecting the Pol($\cdot$) and Inv($\cdot$) operators to pp-definability.

Theorem 2.4 *(Geiger [12]/Bodcharnuk et al. [5]) Let* $\mathbf{B}$ *be a finite relational structure. The set of relations* Inv(Pol($\mathbf{B}$)) *is equal to the set of relations that are pp-definable over* $\mathbf{B}$.

We associate to each algebra $\mathbb{A} = (A, F)$ a set of problems $\mathsf{PPEQ}(\mathbb{A})$, namely, the set containing all problems $\mathsf{PPEQ}(\mathbf{B})$ where $\mathbf{B}$ has universe A and $F \subseteq \mathsf{Pol}(\mathbf{B})$. We define $\mathsf{PPCON}(\mathbb{A})$ and $\mathsf{PPISO}(\mathbb{A})$ similarly. For a complexity class $\mathcal{C}$, we say that the problem $\mathsf{PPEQ}(\mathbb{A})$ is $\mathcal{C}$-hard if $\mathsf{PPEQ}(\mathbb{A})$ contains a problem $\mathsf{PPEQ}(\mathbf{B})$ that is $\mathcal{C}$-hard. We define $\mathcal{C}$-hardness similarly for $\mathsf{PPCON}(\mathbb{A})$ and $\mathsf{PPISO}(\mathbb{A})$.

Theorem 2.5 *Let $\mathbf{B}$ be a finite relational structure, and let $\mathcal{C}$ be a complexity class closed under logspace reduction. The problem $\mathsf{PPEQ}(\mathbf{B})$ is $\mathcal{C}$-hard if and only if $\mathsf{PPEQ}(\mathbb{A}_\mathbf{B})$ is $\mathcal{C}$-hard. The same result holds for $\mathsf{PPCON}(\cdot)$ and $\mathsf{PPISO}(\cdot)$.*

Proof. The proof of [6, Theorem 2] applies to all of the problems. □

The notion of a *variety* is typically defined on indexed algebras; a variety is a class of similar algebras that is closed under the formation of homomorphic images, subalgebras, and products. For our purposes here, however, we may note that the variety generated by an algebra $\mathbb{A}$, denoted by $\mathcal{V}(\mathbb{A})$, is known to be equal to $HSP(\{\mathbb{A}\})$, where the operator H (for instance) is the set of algebras derivable by taking homomorphic images of algebras in the given argument set.

Theorem 2.6 *Suppose that $\mathbb{B} \in \mathcal{V}(\mathbb{A})$. Then, for every problem $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{B})$, there exists a problem $\mathsf{PPEQ}(\mathbf{B}') \in \mathsf{PPEQ}(\mathbb{A})$ such that $\mathsf{PPEQ}(\mathbf{B})$ reduces to $\mathsf{PPEQ}(\mathbf{B}')$, and likewise for $\mathsf{PPCON}(\cdot)$ and $\mathsf{PPISO}(\cdot)$.*

Proof. See Appendix B. □

3 Unary Type

Throughout this section, let $\mathbf{B}$ be a finite relational structure and $\mathcal{V}$ be the variety generated by $\mathbb{A}_\mathbf{B}$. Further assume that $\mathcal{V}$ admits the unary type . In [6] it is shown that if in addition $\mathbb{A}_\mathbf{B}$ is assumed to be idempotent, then $\mathsf{PPEQ}(\mathbf{B})$ is Π_2^p-complete and $\mathsf{PPISO}(\mathbf{B})$ is BOOL-PPISO-hard. In this section we establish the same hardness results, but without assuming the idempotency of $\mathbb{A}_\mathbf{B}$.

Our proof of Theorem 3.1 makes use of the detailed information on tame congruence theory provided in [13] and [15]. This theory associates a *typeset* to a non-trivial finite algebra, which contains one or more of five *types*: (1) the unary type, (2) the affine type, (3) the boolean type, (4) the lattice type, and (5) the semilattice type. By extension, a typeset is associated to each variety, namely, the union of all typesets of finite algebras contained in the variety. A variety is said to *admit* a type if the type is contained in its typeset, and is otherwise said to *omit* the type.

Theorem 3.1 *If $\mathbf{B}$ is a finite relational structure such that the variety generated by $\mathbb{A}_\mathbf{B}$ admits the unary type, then*

- *$\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ are Π_2^p-complete, and*
- *$\mathsf{PPISO}(\mathbf{B})$ is PPISO-complete.*

Prior to embarking on the proof of Theorem 3.1, we point out that from it we can deduce, modulo the G-Set conjecture, dichotomies for the class of pp-equivalence, pp-containment, and pp-isomorphism problems.

Theorem 3.2 *If the G-Set conjecture holds, then for all finite relational structures $\mathbf{B}$, either $\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ are Π_2^p-complete or they are in coNP. In addition, either $\mathsf{PPISO}(\mathbf{B})$ is PPISO-complete and Π_2^p-hard or it is in Σ_2^p.*

It can be remarked that no Π_2^p-hard problem is in Σ_2^p unless $\Pi_2^p = \Sigma_2^p$ and the polynomial hierarchy collapses.

Proof. According to the G-Set conjecture, if $\mathbf{B}$ is a finite relational structure such that the variety generated by $\mathbb{A}_\mathbf{B}$ omits the unary type, then $\mathsf{CSP}(\mathbf{B}^*)$ is in P, where $\mathbf{B}^*$ is obtained from $\mathbf{B}$ by adding to it all relations of the form $\{b\}$ for $b \in B$. From this it follows that $\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ are in coNP and that $\mathsf{PPISO}(\mathbf{B})$ is in Σ_2^p.

On the other hand, if the variety generated by $\mathbb{A}_\mathbf{B}$ admits the unary type, then by Theorem 3.1 we conclude that $\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ are Π_2^p-complete and $\mathsf{PPISO}(\mathbf{B})$ is PPISO-hard; the problem PPISO is straightforwardly verified to be Π_2^p-hard. □

Lemma 3.3 *There is a finite algebra $\mathbb{A}$ in $\mathcal{V}$ and some congruence α on $\mathbb{A}$ such that:*

- *α covers 0_A in $\mathrm{Con}\,(\mathbb{A})$,*
- *the type of the congruence pair $\langle 0_A, \alpha\rangle$ is unary,*
- *the $\langle 0_A, \alpha\rangle$-traces are all polynomially equivalent to two-element sets.*

Proof. This lemma follows from Theorem 6.17 and Lemma 6.18 of [13] along with our assumption that $\mathcal{V}$ admits the unary type. □

Fix an algebra $\mathbb{A}$ and congruence α as in the lemma and choose some $\langle 0_A, \alpha\rangle$-minimal set U, some $\langle 0_A, \alpha\rangle$-trace $N = \{0, 1\}$ contained in U and some unary polynomial $e(x)$ of $\mathbb{A}$ with $e(A) = U$ and $e(x) = e(e(x))$ for all x.

Definition 3.4 (see Definition 6.13 of [13]) *For an element $a \in A$ and $n > 0$, let $\hat{a}_n$ denote the n-tuple $(a, a, \ldots, a)$. We will drop the subscript when the arity of the tuple is clear.*

For an n-ary relation R over N that contains the constant tuples $\hat{0}$ and $\hat{1}$, we define the n-ary relation $\mathbb{A}(R)$ over A to be the universe of the subalgebra of $\mathbb{A}^n$ generated by $R \cup \{\hat{a}_n : a \in A\}$.

Lemma 3.5 *If R is an n-ary relation over N that contains the constant tuples $\hat{0}$ and $\hat{1}$, $M = \{a, b\}$ is a $\langle 0_A, \alpha\rangle$-trace and $p(x)$ is a polynomial of $\mathbb{A}$ with $p(0) = a$ and $p(1) = b$ then $\mathbb{A}(R) \cap M^n = p(R)$. In particular $\mathbb{A}(R) \cap N^n = R$.*

Furthermore, any member $(a_1, \ldots, a_n)$ of $\mathbb{A}(R)$ is α-constant, i.e., $(a_i, a_j) \in \alpha$ for all $1 \leq i \leq j \leq n$ and if $p(x)$ is a unary polynomial of $\mathbb{A}$, then $(p(a_1), \ldots, p(a_n)) \in \mathbb{A}(R)$.

Proof. The first part follows from Lemma 6.14 (2) of [13] and from the fact that any two $\langle 0_A, \alpha\rangle$-traces are polynomially isomorphic in $\mathbb{A}$ (see Corollary 5.2. (2) of [13]). The second holds since N is contained in a single α-class of $\mathbb{A}$ and $\mathbb{A}(R)$ contains all constant tuples and is a subuniverse of $\mathbb{A}^n$. □

In the proof of the hardness results of this section, we will need detailed information about relations over A of the form $\mathbb{A}(R)$, where R is a relation over N. We employ the theory of multitraces, developed in [15], to aid us. For a more general presentation of this notion, see Section 3 of that article.

Definition 3.6 *A multitrace is a subset T of A of the form $f(N, N, \ldots, N)$ for some polynomial $f(\bar{x})$ of $\mathbb{A}$.*

We can use the notion of a multitrace to gain a clear picture of the relations $\mathbb{A}(R)$ in the special case where $n > 0$ and $R = N^n$. For the rest of this section, let $k = |A|$.

Proposition 3.7 *For $n > 0$, let $\tau_n = \mathbb{A}(N^n)$.*

- $\tau_n = \bigcup\{T^n : T \text{ is a multitrace}\}$.
- *For $n \geq k$, we have*

$$\tau_n(x_1, \ldots, x_n) \text{ iff } \bigwedge_{\{1 \leq i_1 < \cdots < i_k \leq n\}} \tau_k(x_{i_1}, \ldots, x_{i_k}).$$

Proof. If T is a multitrace, then there is some m-ary polynomial $f(x_1, \ldots, x_m)$ of $\mathbb{A}$ such that $T = f(N, N, \ldots, N)$. Since τ_n is the universe of the subalgebra of $\mathbb{A}^n$ generated by N^n and all of the constant n-tuples, it follows that $T^n \subseteq \tau_n$. Conversely, if $\sigma = (a_1, \ldots, a_n)$ is in τ_n then there is some $m > 0$, some m-ary polynomial $t(\bar{x})$ of $\mathbb{A}$ and n-tuples $b_i \in N^n$, for $1 \leq i \leq m$, such that $\sigma = t(b_1, \ldots, b_m)$ (where t is applied coordinatewise). But then $\sigma \in T^n$, where T is the multitrace $t(N, N, \ldots, N)$.

For the second part of this proposition, suppose that $n > k$ and $\sigma = (a_1, \ldots, a_n)$ is in τ_n. Then there is some multitrace T with $a_i \in T$ for all $1 \leq i \leq n$. In particular, if $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ then $a_{i_j} \in T$ for all $1 \leq j \leq k$ and from this it follows that $(a_{i_1}, \ldots, a_{i_k}) \in \tau_k$.

In the other direction, since $k = |A|$, we can choose some sequence $1 \leq i_i < i_2 < \cdots < i_k \leq n$ such that for all $1 \leq j \leq n$, $a_j = a_{i_m}$ for some $m \leq k$. If $(a_{i_1}, \ldots, a_{i_k}) \in \tau_k$ then by the first part of this proposition, there is some multitrace T with $(a_{i_1}, \ldots, a_{i_k}) \in T^k$ and thus $\sigma \in T^n$. □

The following theorem records some relevant features of multitraces.

Theorem 3.8 *Let T be a multitrace, say $T = f(N, N, \ldots, N)$ for some m-ary polynomial f of $\mathbb{A}$. There is a p-ary polynomial $f'(\bar{x})$ of $\mathbb{A}$ for some $p \leq m$ and some unary polynomials (called coordinate maps) $g_i(x)$ of $\mathbb{A}$, for $1 \leq i \leq p$, such that*

- *$T = f'(N, N, \ldots, N)$ and $g_i(T) \subseteq N$ for all i,*
- *for all $x_j \in N$, and all i, $g_i(f'(x_1, \ldots, x_p)) = x_i$,*
- *for all $x \in T$, $x = f'(g_1(x), \ldots, g_p(x))$,*
- *the set N^p is in bijective correspondence with T via the map that takes a p-tuple $(n_1, \ldots, n_p)$ to $f'(n_1, \ldots, n_p)$.*

Proof. This follows from Theorem 3.10 of [15]. □

The following definition provides a way to translate primitive positive formulas over a set of Boolean relations to primitive positive formulas over relations compatible with $\mathbb{A}$. This translation will be used to establish our hardness results.

Definition 3.9 *Let $\mathcal{R}$ be a set of finitary relations over $\{0, 1\}$. If $\phi(x_1, \ldots, x_n)$ is a pp-formula over the relations in $\mathcal{R}$ and if $\{y_1, \ldots, y_m\}$ is its set of quantified variables, define $\mathbb{A}(\phi)(x_1, \ldots, x_n)$ to be the pp-formula over the relations $\{\mathbb{A}(R) : R \in \mathcal{R}\} \cup \{\tau_j : j \leq k\}$ obtained from ϕ by replacing each occurrence of a relation $R \in \mathcal{R}$ by $\mathbb{A}(R)$ and by conjoining the formula $\tau_{n+m}(x_1, \ldots, x_n, y_1, \ldots, y_m)$. If $n + m > k$, we make use of the second part of Proposition 3.7 to express τ_{n+m} in terms of τ_k.*

Theorem 3.10 *If $\phi(x_1, \ldots, x_n)$ is a pp-formula over the set of finitary relations $\mathcal{R}$ over $\{0, 1\}$ and each $R \in \mathcal{R}$ contains the constant tuples $\hat{0}$ and $\hat{1}$ then*

$$\{\mu \in \{0, 1\}^n : \phi(\mu)\} = \{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} \cap \{0, 1\}^n$$

and

$$\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} = \mathbb{A}(\{\mu \in \{0, 1\}^n : \phi(\mu)\}).$$

Proof. Suppose that $\{y_1, \ldots, y_m\}$ are the quantified variables of ϕ. For the first equality, if $\mu = (a_1, \ldots, a_n) \in \{0, 1\}^n$ and $\phi(\mu)$ is witnessed by the elements $b_i \in \{0, 1\}$, $1 \leq i \leq m$, then all of these elements are in N and hence $\tau_{n+m}(a_1, \ldots, a_n, b_1, \ldots, b_m)$ holds. If some clause $R(u_1, \ldots, u_m)$ of ϕ holds,

where $u_i \in \{a_1, \ldots, a_n, b_1, \ldots, b_m\}$ then by construction $\mathbb{A}(R)(u_1, \ldots, u_n)$ also holds. From this it follows that $\mathbb{A}(\phi)(a_1, \ldots, a_n)$ holds, using the same witnesses b_i, $1 \leq i \leq m$.

Conversely, suppose that $\sigma = (a_1, \ldots, a_n) \in \{0,1\}^n$ and $\mathbb{A}(\phi)(\sigma)$ holds, witnessed by the elements $b_i \in A$, $1 \leq i \leq m$. Since each relation $\mathbb{A}(R)$ that appears as a clause in $\mathbb{A}(\phi)$ is closed under the unary operation $e(x)$, applied coordinatewise (see Lemma 3.5), it follows that the elements $e(b_i)$, $1 \leq i \leq m$ also witness that $\mathbb{A}(\phi)(\sigma)$ holds. We use here that e is the identity map on its range, and in particular that $e(0) = 0$ and $e(1) = 1$. We claim that in fact, the elements $e(b_j)$ are all members of $\{0,1\}$. Since the clause $\tau_{n+m}(a_1, \ldots, a_n, b_1, \ldots, b_m)$ holds then all of these elements lie in the same α-class, and in particular belong to the α-class that contains 0, since $a_1 \in \{0,1\}$. We conclude that for each j, $e(b_j) \in \{0,1\}$ since e maps the entire α-class that contains 0 into $\{0,1\}$.

Finally, Lemma 3.5 provides that $\mathbb{A}(R) \cap \{0,1\}^n = R$ for all $R \in \mathcal{R}$ and from this it follows that the elements $e(b_j)$, $1 \leq j \leq m$, also witness that $\phi(\sigma)$ holds. Thus the first equality has been established.

For the second equality, we can apply the $\mathbb{A}(\cdot)$ operator to both sides of the first equality to obtain that $\mathbb{A}(\{\mu \in \{0,1\}^n : \phi(\mu)\})$ is equal to

$$\mathbb{A}(\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} \cap \{0,1\}^n)$$

and so it will suffice to prove that

$$\mathbb{A}(\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} \cap \{0,1\}^n) = \{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\}$$

to complete the proof.

The containment of the left hand side of this derived equality in the set $\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\}$ follows after observing that this set is a subuniverse of $\mathbb{A}^n$ and that it contains all constant n-tuples $\hat{c}$, for $c \in A$. The latter follows from the fact that each relation $\mathbb{A}(R)$ contains all constant tuples and so every constant n-tuple over A is a solution of $\mathbb{A}(\phi)$.

For the remaining containment, let $\sigma = (a_1, \ldots, a_n)$ be a solution of $\mathbb{A}(\phi)$, witnessed by the elements $b_i \in A$, $1 \leq i \leq m$. Since $\tau_{n+m}(a_1, \ldots, a_n, b_1, \ldots, b_m)$ holds then there is some multitrace T of $\mathbb{A}$ that contains all of these elements. By Theorem 3.8, it follows that for some $p > 0$, there is some p-ary polynomial f' and coordinate maps $g_i(x)$, $1 \leq i \leq p$, that satisfy the properties stated in that theorem. In particular, $T = f'(N, N, \ldots, N)$ and for all $c \in T$, $c = f'(g_1(c), \ldots, g_p(c))$.

From Lemma 3.5 it follows that for any q-ary relation $R \in \mathcal{R}$, if $c_j \in T$, for $1 \leq j \leq q$, and $(c_1, \ldots, c_q) \in \mathbb{A}(R)$, then

$$(g_i(c_1), \ldots, g_i(c_q)) \in (\mathbb{A}(R) \cap \{0,1\}^q) = R,$$

for any i. This is because for each coordinate map g_i, we have that $g_i(T) \subseteq \{0,1\}$. Extending this to our pp-formula $\mathbb{A}(\phi)$, it follows that not only is $(g_i(a_1), \ldots, g_i(a_n))$ a solution, witnessed by $g_i(b_j)$, $1 \leq j \leq m$, but it is also a member of $\{0,1\}^n$. So, each of these tuples belongs to the generating set of the relation $\mathbb{A}(\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} \cap \{0,1\}^n)$.

Since for each j, $a_j = f'(g_1(a_j), \ldots, g_p(a_j))$ and $\mathbb{A}(\{\sigma \in A^n : \mathbb{A}(\phi)(\sigma)\} \cap \{0,1\}^n)$ is closed under f', applied coordinatewise, we conclude that $\sigma = (a_1, \ldots, a_n)$ is also a member of this relation, as required. □

Corollary 3.11 *Let $\mathcal{R}$ be a set of finitary relations over $\{0,1\}$ such that each $R \in \mathcal{R}$ contains the constant tuples. Two pp-formulas $\phi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ over the relations in $\mathcal{R}$ will be equivalent (contained, isomorphic with respect to sectionings) if and only if the pp-formulas $\mathbb{A}(\phi)$ and $\mathbb{A}(\psi)$ are equivalent (contained, isomorphic with respect to sectionings) over $\{\mathbb{A}(R) : R \in \mathcal{R}\} \cup \{\tau_j : j \leq k\}$.*

Proof of Theorem 3.1. To establish that $\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ are Π_2^p-complete, it will suffice to show that it is Π_2^p-hard, since both problems are contained in Π_2^p. By Theorem 4 and Lemma 4 from [6] it follows that there is some finite relational structure $\mathbf{C} = (\{0,1\}, \mathcal{R})$ such that each $R \in \mathcal{R}$ contains the constant tuples $\hat{0}$ and $\hat{1}$ and such that $\mathsf{PPEQ}(\mathbf{C})$ and $\mathsf{PPCON}(\mathbf{C})$ are Π_2^p-hard. By those proofs, one also readily obtains that $\mathsf{PPISO}(\mathbf{C})$ is BOOL-PPISO-hard (for the definition of BOOL-PPISO given in this paper).

From Corollary 3.11 there is a finite algebra $\mathbb{A}$ in the variety generated by $\mathbb{A}_\mathbf{B}$ such that $\mathsf{PPEQ}(\mathbf{C})$, $\mathsf{PPCON}(\mathbf{C})$ and $\mathsf{PPISO}(\mathbf{C})$ reduce to $\mathsf{PPEQ}(\mathbb{A})$, $\mathsf{PPCON}(\mathbb{A})$, and $\mathsf{PPISO}(\mathbb{A})$ respectively. Finally, using Theorems 2.6 and 2.5 we conclude that $\mathsf{PPEQ}(\mathbf{B})$ and $\mathsf{PPCON}(\mathbf{B})$ is Π_2^p-complete and $\mathsf{PPISO}(\mathbf{B})$ is BOOL-PPISO-hard; by Proposition 2.2, $\mathsf{PPISO}(\mathbf{B})$ is PPISO-hard and hence PPISO-complete. □

4 Non-Congruence Modularity

A variety is *idempotent* if each algebra in the variety is idempotent. A variety is *congruence modular* if the congruence lattice of each algebra in the variety is modular. We let MFISO denote the problem of deciding whether two monotone Boolean formulas are isomorphic; recall that a monotone Boolean formula is one formed using just the connectives AND ($\wedge$) and OR ($\vee$).

Theorem 4.1 *Let $\mathbf{B}$ be a finite relational structure. If $\mathcal{V}(\mathbb{A}_\mathbf{B})$ is idempotent and not congruence modular, then:*

(i) $\mathsf{PPEQ}(\mathbf{B})$ *and* $\mathsf{PPCON}(\mathbf{B})$ *are coNP-hard, and*

(ii) $\mathsf{PPISO}(\mathbf{B})$ *is* MFISO*-hard.*

The Edinburgh conjecture, discussed in the introduction, predicts that for all structures $\mathbf{B}$, either $\mathcal{V}(\mathbb{A}_\mathbf{B})$ is not congruence modular, or $\mathbb{A}_\mathbf{B}$ has few subpowers. This conjecture would imply a P/coNP-hard dichotomy for the $\mathsf{PPEQ}(\cdot)$ and $\mathsf{PPCON}(\cdot)$ problems for all such structures: by the conjecture, the structures not covered by Theorem 4.1 have that $\mathbb{A}_\mathbf{B}$ has few subpowers, in which case the problem $\mathsf{PPEQ}(\mathbf{B})$ is in P by [6, Theorem 7], and $\mathsf{PPCON}(\mathbf{B})$ is in P as well by an application of Proposition 2.3. Similarly, under this conjecture we have a dichotomy in the complexity of $\mathsf{PPISO}(\mathbf{B})$: either we have MFISO-hardness by the theorem–and hence coNP-hardness by [19, Theorem 19]–or, we have that $\mathsf{PPISO}(\mathbf{B})$ is in NP as a consequence of $\mathsf{PPEQ}(\mathbf{B})$ being in P. Note that this would give a dichotomy unless the polynomial hierarchy collapses, as no coNP-hard problem can be in NP unless NP = coNP.

Proof. We exploit the fact, enlightened by [17, Lemma 1 and Lemma 2], that the failure of congruence modularity in $\mathcal{V}(\mathbb{A}_\mathbf{B})$ is nicely witnessed by a finite algebra $\mathbb{A} \in \mathcal{V}(\mathbb{A}_\mathbf{B})$, and congruences α, β, and γ of $\mathbb{A}$ with the following properties: there exist finite algebras $\mathbb{B}$ and $\mathbb{C}$ in $\mathcal{V}(\mathbb{A}_\mathbf{B})$ such that $\mathbb{A} = \mathbb{B} \times \mathbb{C}$; β and γ are the kernels of the projections of $\mathbb{A}$ onto $\mathbb{B}$ and $\mathbb{C}$ respectively; there exist a partition of B into two nonempty blocks B_0 and B_1, and congruences $\alpha_0 < 1_\mathbb{C}$ and $\alpha_1 = 1_\mathbb{C}$ of $\mathbb{C}$ such that α is

$$\{((b,c),(b,c')) \mid b \in B_i \Rightarrow (c,c') \in \alpha_i, i = 0,1\}. \quad (1)$$

Note that β and γ form a pair of factor congruences on $\mathbb{A}$ and $\alpha < \beta$; thus, $0_\mathbb{A}$, α, β, γ, and $1_\mathbb{A}$ form a pentagon in the congruence lattice of $\mathbb{A}$, thus witnessing the failure of congruence modularity in $\mathcal{V}(\mathbb{A}_\mathbf{B})$.

Let $\mathbf{A}$ be the relational structure with universe $A = B \times C$ and relations α, β, and γ.

(i) We now prove that $\mathsf{PPCON}(\mathbf{A})$ is coNP-hard. Note that $\mathsf{PPCON}(\mathbf{A})$ is in $\mathsf{PPCON}(\mathbb{A})$, because α, β, and γ are congruences of $\mathbb{A}$, and hence are preserved by the fundamental operations of $\mathbb{A}$. Since $\mathbb{A} \in \mathcal{V}(\mathbb{A}_\mathbf{B})$, by Theorem 2.6, $\mathsf{PPCON}(\mathbf{A})$ logspace reduces to some $\mathsf{PPCON}(\mathbf{B}')$ in $\mathsf{PPCON}(\mathbb{A}_\mathbf{B})$, so that $\mathsf{PPCON}(\mathbb{A}_\mathbf{B})$ is coNP-hard, which finally implies that $\mathsf{PPCON}(\mathbf{B})$ is coNP-hard by Theorem 2.5. The coNP-hardness of $\mathsf{PPEQ}(\mathbf{B})$ follows immediately by Proposition 2.3.

We describe a reduction from the coNP-complete problem of deciding entailment between two monotone Boolean formulas [4, Theorem 4.1], call it MBCON, to $\mathsf{PPCON}(\mathbf{A})$. The reduction has two stages, which we now prepare.

We introduce the following intermediate problem. Let B and C be disjoint finite sets. Let $\mathbf{S}$ be a *sorted* structure, that is, a relational structure with universe $B \cup C$ and *sorts* B and C, equipped with the ternary *sorted* relation $R \subseteq B \times C \times C$ given by,

$$\{(b,c,c') \mid b \in B_i \Rightarrow (c,c') \in \alpha_i, i = 0,1\}. \quad (2)$$

A *sorted* pp-formula ϕ on $\mathbf{S}$ is a conjunction of constraints of either the form $R(x,y,z)$, with x of sort B and y, z of sort C, or the form $x = y$, with x and y of the same sort, where some variables can be existentially quantified. A *sorted* assignment sends variables of sort B to B, and variables of sort C to C. A sorted assignment f of the free variables of ϕ *satisfies* ϕ over $\mathbf{S}$ if there exists a sorted assignment of all the variables of ϕ, extending f, that satisfies each constraint of ϕ. Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ be variables of sort B and C respectively, and let ϕ and ψ be sorted pp-formulas over $\mathbf{S}$ having free variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$. Then, ϕ *entails* ψ over $\mathbf{S}$, if and only if the satisfying assignments of ψ over $\mathbf{S}$ contain the satisfying assignments of ϕ over $\mathbf{S}$; the *entailment problem on* $\mathbf{S}$ is to decide, given two sorted pp-formulas ϕ and ψ as above, whether or not ϕ entails ψ over $\mathbf{S}$.

The proof proceeds by first reducing MBCON to the entailment problem on $\mathbf{S}$, and then reducing the entailment problem on $\mathbf{S}$ to $\mathsf{PPCON}(\mathbf{A})$.

The reduction from MBCON to the entailment problem on $\mathbf{S}$ works as follows. Let $\mathbf{x} = x_1, \ldots, x_n$, and let $\phi(\mathbf{x})$ be a monotone Boolean formula. By induction on the structure of $\phi(\mathbf{x})$, we construct a sorted pp-formula $\phi'(\mathbf{x}, y_1, y_2)$ on $\mathbf{S}$, where the variables $\mathbf{x}$ are of sort B and the variables y_1, y_2 are of sort C, as follows: for $i \in \{1, \ldots, n\}$, if $\phi(\mathbf{x})$ is x_i, then $\phi'(\mathbf{x}, y_1, y_2) = R(x_i, y_1, y_2)$; if $\phi(\mathbf{x})$ is $\phi_1(\mathbf{x}) \wedge \phi_2(\mathbf{x})$, then $\phi'(\mathbf{x}, y_1, y_2) = \phi'_1(\mathbf{x}, y_1, y_2) \wedge \phi'_2(\mathbf{x}, y_1, y_2)$; if $\phi(\mathbf{x})$ is $\phi_1(\mathbf{x}) \vee \phi_2(\mathbf{x})$, then

$$\phi' = (\exists z)\,(\phi'_1(\mathbf{x}, y_1, z) \wedge \phi'_2(\mathbf{x}, z, y_2))\,, \quad (3)$$

where z is a fresh variable of sort C. The reduction is now, given an instance (ϕ, ψ) of MBCON, construct the instance (ϕ', ψ') of the entailment problem on $\mathbf{S}$.

We now prove that the reduction is correct. The key step is to establish a correspondence between the satisfying assignments of ϕ (over the two-element Boolean lattice) and ϕ' over $\mathbf{S}$. Let f be a Boolean assignment of variables $x_1, \ldots, x_n$, and let g be a sorted assignment of variables $x_1, \ldots, x_n, y_1, y_2$. Say that f and g *match* if: $f(x_i) = 0$ implies $g(x_i) \in B_0$, and $f(x_i) = 1$ implies $g(x_i) \in B_1$.

Claim 4.2 *Let g be a sorted assignment of $x_1, \ldots, x_n, y_1, y_2$, matching the Boolean assignment f of $x_1, \ldots, x_n$. Then, g satisfies ϕ' over $\mathbf{S}$ if and only if, f does not satisfy ϕ implies $(g(y_1), g(y_2)) \in \alpha_0$.*

Proof. ($\Rightarrow$) Suppose that g satisfies ϕ' over $\mathbf{S}$. The proof is by induction on the structure of ϕ'. If $\phi' = R(x_i, y_1, y_2)$, then by (2) $g(x_i) \in B_j$ implies $(g(y_1), g(y_2)) \in \alpha_j$ for

$j = 0, 1$. By construction, $\phi = x_i$, and if f does not satisfy ϕ, then $g(x_i) \in B_0$ (as f matches g), and then $(g(y_1), g(y_2)) \in \alpha_0$. If $\phi' = \phi'_1 \wedge \phi'_2$, then g satisfies both ϕ'_1 and ϕ'_2 over $\mathbf{S}$. By construction, $\phi = \phi_1 \wedge \phi_2$. If f does not satisfy ϕ, then either f does not satisfy ϕ_1 or f does not satisfy ϕ_2; say without loss of generality that f does not satisfy ϕ_1. By the induction hypothesis, $(g(y_1), g(y_2)) \in \alpha_0$. If $\phi'(\mathbf{x}, y_1, y_2) = (\exists z)(\phi'_1(\mathbf{x}, y_1, z) \wedge \phi'_2(\mathbf{x}, z, y_2))$, then there exists a point $c \in C$ such that extending g by $g(z) = c$, g satisfies both $\phi'_1(\mathbf{x}, y_1, z)$ and $\phi'_2(\mathbf{x}, z, y_2)$ over $\mathbf{S}$. By construction, $\phi = \phi_1 \vee \phi_2$. If f does not satisfy ϕ, then f does not satisfy ϕ_1 and f does not satisfy ϕ_2. Then, by the induction hypothesis, $(g(y_1), g(z)) \in \alpha_0$ and $(g(z), g(y_2)) \in \alpha_0$, so that by transitivity, $(g(y_1), g(y_2)) \in \alpha_0$.

($\Leftarrow$) Suppose that g does not satisfy ϕ' over $\mathbf{S}$. We show that f does not satisfy ϕ but $(g(y_1), g(y_2)) \notin \alpha_0$. The proof is by induction on the structure of ϕ'. If $\phi' = R(x_i, y_1, y_2)$, then by (2) the only possibility for g to not satisfy ϕ' is when $g(x_i) \in B_0$ and $(g(y_1), g(y_2)) \notin \alpha_0$ (note that in fact, $g(x_i) \in B_1$ implies $(g(y_1), g(y_2)) \in \alpha_1$ holds trivially because $\alpha_1 = 1_{\mathbb{C}}$). By construction, $\phi = x_i$. As f matches g, by definition $f(x_i) = 0$, that is, f does not satisfy ϕ, and we are done. If $\phi' = \phi'_1 \wedge \phi'_2$, then either g does not satisfy ϕ'_1 over $\mathbf{S}$, or g does not satisfy ϕ'_2 over $\mathbf{S}$; say without loss of generality that g does not satisfy ϕ'_1 over $\mathbf{S}$. By construction, $\phi = \phi_1 \wedge \phi_2$, so by the induction hypothesis, f does not satisfy ϕ_1 but $(g(y_1), g(y_2)) \notin \alpha_0$. It follows that f does not satisfy ϕ, and we are done. Finally, let $\phi'(\mathbf{x}, y_1, y_2) = (\exists z)(\phi'_1(\mathbf{x}, y_1, z) \wedge \phi'_2(\mathbf{x}, z, y_2))$; recall that in this case, by construction, $\phi = \phi_1 \vee \phi_2$. If g does not satisfy ϕ' over $\mathbf{S}$, then there does not exist a point $c \in C$ such that extending g by $g(z) = c$, g satisfies both $\phi'_1(\mathbf{x}, y_1, z)$ and $\phi'_2(\mathbf{x}, z, y_2)$ over $\mathbf{S}$. In particular, the extension $g(z) = g(y_1)$ of g satisfies $\phi'_1(\mathbf{x}, y_1, z)$ over $\mathbf{S}$, hence it does not satisfy $\phi'_2(\mathbf{x}, z, y_2)$ over $\mathbf{S}$; here, the induction hypothesis gives that f does not satisfy ϕ_2, but $(g(z), g(y_2)) = (g(y_1), g(y_2)) \notin \alpha_0$. Similarly, the extension $g(z) = g(y_2)$ of g satisfies $\phi'_2(\mathbf{x}, z, y_2)$ over $\mathbf{S}$, hence it does not satisfy $\phi'_1(\mathbf{x}, y_1, z)$ over $\mathbf{S}$, and the induction hypothesis gives that f does not satisfy ϕ_1, but $(g(y_1), g(z)) = (g(y_1), g(y_2)) \notin \alpha_0$. Hence, f does not satisfy ϕ, but $(g(y_1), g(z)) = (g(y_1), g(y_2)) \notin \alpha_0$, and we are done. □

Let ϕ and ψ be Boolean formulas, and let ϕ' and ψ' be the sorted pp-formulas given by the reduction. Note that Boolean assignments of $x_1, \ldots, x_n$ determine a partition of sorted assignments of $x_1, \ldots, x_n, y_1, y_2$: for, sorted assignments g and g' are in the same block of the partition if and only if they match the same Boolean assignment f. By Claim 4.2, a sorted assignment g satisfies ϕ' (respectively, ψ') if and only if its matching Boolean assignment satisfies ϕ (respectively, ψ), or $(g(y_1), g(y_2)) \in \alpha_0$; therefore, the Boolean assignments satisfying ψ contain the Boolean assignments satisfying ϕ if and only if, the sorted assignments satisfying ψ' over $\mathbf{S}$ contain the sorted assignments satisfying ϕ' over $\mathbf{S}$.

This completes the first part of the proof. The second reduction, from the entailment problem on $\mathbf{S}$ to $\mathsf{PPCON}(\mathbf{A})$, works as follows. Let ϕ be a sorted pp-formula on $\mathbf{S}$ over variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ of sort B and C respectively. We construct a pp-formula ϕ' on $\mathbf{A}$, as follows. For each variable z in ϕ, introduce a fresh variable $z' = (z_1, z_2)$; z' is existentially quantified in ϕ' if and only if z is existentially quantified in ϕ'. If ϕ contains the constraint $x_i = x_j$ for some $i, j \in \{1, \ldots, n\}$, then ϕ' contains the conjunct $\beta(x'_i, x'_j)$; if ϕ contains the constraint $y_i = y_j$ for some $i, j \in \{1, \ldots, m\}$, then ϕ' contains the conjunct $\gamma(y'_i, y'_j)$; if ϕ contains the constraint $R(x_i, y_j, y_k)$, then ϕ' contains the conjunct

$$(\exists w'_1)(\exists w'_2)(\beta(w'_1, x'_i) \wedge \beta(w'_2, x'_i) \wedge \gamma(w'_1, y'_j) \wedge \gamma(w'_2, y'_k) \wedge \alpha(w'_1, w'_2)), \tag{4}$$

where w'_1 and w'_2 are fresh variables. The reduction is now, given an instance (ϕ, ψ) of the entailment problem on $\mathbf{S}$, construct the instance (ϕ', ψ') of $\mathsf{PPCON}(\mathbf{A})$. The construction is a sequence of local substitutions, hence it is feasible in logspace.

We now prove that the reduction is correct. The key step is to establish a correspondence between the satisfying assignments of ϕ over $\mathbf{S}$ and ϕ' over $\mathbf{A}$. Let f be a sorted assignment of variables $x_1, \ldots, x_n$ in B and $y_1, \ldots, y_m$ in C, and let g be an assignment of variables $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m$ in $A = B \times C$. Say that f and g *match* if: $f(x_i) = b$ implies $g(x'_i) = (b, \cdot)$, and $f(y_i) = c$ implies $g(y'_i) = (\cdot, c)$. For sake of notation, say that the free variables in ϕ are $x'_1, \ldots, x'_{n'}, y'_1, \ldots, y'_{m'}$.

Claim 4.3 *Let g be an assignment of $x'_1, \ldots, x'_{n'}, y'_1, \ldots, y'_{m'}$ in $B \times C$, matching the sorted assignment f of $x_1, \ldots, x_{n'}, y_1, \ldots, y_{m'}$. Then, g satisfies ϕ' over $\mathbf{A}$ if and only if f satisfies ϕ over $\mathbf{S}$.*

Proof. ($\Rightarrow$) Suppose that g satisfies ϕ' over $\mathbf{A}$, and let g' be an extension of g to the quantified variables of ϕ' that satisfies each conjunct in ϕ'. Let f' be the sorted assignment of the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ in ϕ defined by $f'(x_i) = b$ if and only if $g'(x'_i) = (b, \cdot)$, and $f'(y_i) = c$ if and only if $g'(y'_i) = (\cdot, c)$; by definition, the restriction of f' to the free variables of ϕ, that is, f, matches g. We prove that f satisfies ϕ over $\mathbf{S}$, by checking that f' satisfies each conjunct of ϕ over $\mathbf{S}$.

If g' satisfies the conjunct $\beta(x'_i, x'_j)$ in ϕ', that is, $g'(x'_i) = (b, \cdot)$ and $g'(x'_j) = (b, \cdot)$ for some $b \in B$, then f' satisfies the counterpart $x_i = x_j$ in ϕ because $f'(x_i) = f'(x_j) = b$ by definition of f'. The case of conjuncts of the

form $\gamma(y'_i, y'_j)$ in ϕ' is similar. If g' satisfies a conjunct of the form in (4) in ϕ', then by direct inspection, the following holds: $g'(x'_i) = (b, \cdot)$, $g'(w'_1) = (b, \cdot)$, and $g'(w'_2) = (b, \cdot)$ for some $b \in B$; $g'(y'_j) = (\cdot, c_1)$ and $g'(w'_1) = (\cdot, c_1)$ for some $c_1 \in C$; $g'(y'_k) = (\cdot, c_2)$ and $g'(w'_2) = (\cdot, c_2)$ for some $c_2 \in C$; and, $((b, c_1), (b, c_2)) \in \alpha$. By (1), this implies that, $b \in B_i$ implies $(c_1, c_2) \in \alpha_i$ for $i = 1, 2$. But then, by (2), f' satisfies the counterpart in ϕ of the conjunct under consideration, $R(x_i, y_j, y_k)$, because by definition, $f'(x_i) = b$, $f'(y_j) = c_1$, and $f'(y_k) = c_2$.

($\Leftarrow$) Conversely, suppose that the sorted assignment f satisfies ϕ over $\mathbf{S}$. Let f' be an extension of f to the quantified variables of ϕ that satisfies each constraint in ϕ. Let g' be the assignment of the variables $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m$ in ϕ onto $B \times C$ defined by $g'(x'_i) = (b, c_i)$ if and only if $f'(x_i) = b$ and $g'(y'_i) = (b_i, c)$ if and only if $f'(y_i) = c$, where b_i and c_i are arbitrary fixed points in B and C respectively. By definition, the restriction of g' to the free variables of ϕ', call it g, matches f. We prove that g satisfies ϕ' over $\mathbf{A}$, by checking that g' satisfies each conjunct of ϕ' over $\mathbf{A}$.

If f' satisfies the constraint $x_i = x_j$ in ϕ, that is, $f'(x_i) = f'(x_j) = b$ for some $b \in B$, then g' satisfies the counterpart $\beta(x'_i, x'_j)$ in ϕ', because $g'(x'_i) = (b, \cdot)$ and $g'(x'_j) = (b, \cdot)$ by definition of g'. The case of constraints of the form $y_i = y_j$ in ϕ is similar. If f' satisfies a constraint of the form $R(x_i, y_j, y_k)$ in ϕ, then by (2), $f(x_i) = b \in B_l$ implies $(f'(y_j), f'(y_k)) = (c_1, c_2) \in \alpha_l$ for $l = 1, 2$. In this case, a conjunct ζ of the form in (4) occurs in ϕ'. We extend g' to the existentially quantified variables w'_1 and w'_2 in (4) by letting $g'(w'_1) = (b, c_1)$ and $g'(w'_2) = (b, c_2)$. By direct inspection, this extension of g' satisfies ζ. $\square$

Now, let ϕ and ψ be sorted pp-formulas on $\mathbf{S}$, and let ϕ' and ψ' be the pp-formulas on $\mathbf{A}$ given by the reduction. Note that sorted assignments of $x_1, \ldots, x_{n'}, y_1, \ldots, y_{m'}$ determine a partition of the assignments of $x'_1, \ldots, x'_{n'}, y'_1, \ldots, y'_{m'}$ in $B \times C$: for, assignments g and g' are in the same block of the partition if and only if they match the same sorted assignment f. By Claim 4.3, the satisfying assignments of ϕ' (respectively, ψ') are exactly those in blocks that match satisfying assignments of ϕ (respectively, ψ). Hence, the sorted assignments satisfying ϕ over $\mathbf{S}$ are contained in the sorted assignments satisfying ψ over $\mathbf{S}$, if and only if the assignments satisfying ϕ over $\mathbf{A}$ are contained in the assignments satisfying ψ over $\mathbf{A}$. This completes the second part of the proof.

We conclude that $\mathsf{PPCON}(\mathbf{A})$ is coNP-hard, and the proof of statement (i) is complete.

(ii) We now prove that $\mathsf{PPISO}(\mathbf{A})$ is MFISO-hard. Along the lines of the first paragraph of part (i), noticing that $\mathsf{PPISO}(\mathbf{A})$ is in $\mathsf{PPISO}(\mathbb{A})$, this implies that $\mathsf{PPEQ}(\mathbf{B})$ is MFISO-hard.

We describe a reduction from the problem of deciding whether two monotone Boolean formulas are isomorphic to the problem $\mathsf{PPISO}(\mathbf{A})$. Along the lines of part (i), the reduction proceeds in two stages. Say that two sorted pp-formulas are *isomorphic (on* $\mathbf{S}$*)* if and only if they have an isomorphism on $\mathbf{S}$ that fixes the sorts, that is, sends variables of sort B (respectively, C) to variables of sort B (respectively, C). The *isomorphism problem on* $\mathbf{S}$ is to decide, given two sorted pp-formulas ϕ and ψ, whether or not ϕ and ψ are isomorphic over $\mathbf{S}$.

First, we reduce from the monotone Boolean formulas isomorphism problem to $\mathsf{PPISO}(\mathbf{S})$. Let (ϕ, ψ) be a pair of monotone Boolean formulas over variables $x_1, \ldots, x_n$, and let (ϕ', ψ') be the pair of sorted pp-formulas on $\mathbf{S}$, over variables $x_1, \ldots, x_n$ of sort B and variables y_1, y_2 of sort C, computed in part (i). The following claim shows that the reduction is correct.

Claim 4.4 *ϕ and ψ are isomorphic if and only if ϕ' and ψ' are isomorphic.*

Proof. ($\Rightarrow$) Let π be an isomorphism of ϕ and ψ, and let $\pi'(x_i) = \pi(x_i)$ and $\pi'(y_j) = y_j$ for $i = 1, \ldots, n$ and $j = 1, 2$. We claim that π' is an isomorphism of ϕ' and ψ' on $\mathbf{S}$ (clearly, π' fixes the sorts). Let g be a sorted assignment of $x_1, \ldots, x_n, y_1, y_2$, and let f be the Boolean assignment matching g. First suppose that g satisfies ϕ' on $\mathbf{S}$. By Claim 4.2, either $(g(y_1), g(y_2)) \in \alpha_0$, or f satisfies ϕ. In the former case, $(g \circ \pi'(y_1), g \circ \pi'(y_2)) = (g(y_1), g(y_2)) \in \alpha_0$, and $g \circ \pi'$ satisfies ψ' on $\mathbf{S}$ by Claim 4.2. In the latter case, by hypothesis, $f \circ \pi$ satisfies ψ. Since $g \circ \pi'$ and $f \circ \pi$ match, by Claim 4.2, $g \circ \pi'$ satisfies ψ' on $\mathbf{S}$. Conversely, suppose that g does not satisfy ϕ' on $\mathbf{S}$. By Claim 4.2, f does not satisfy ϕ and $(g(y_1), g(y_2)) \notin \alpha_0$. By hypothesis, $f \circ \pi$ does not satisfy ψ. As $g \circ \pi'$ and $f \circ \pi$ match, by Claim 4.2, $g \circ \pi'$ does not satisfy ψ' on $\mathbf{S}$.

($\Leftarrow$) Let π' witness that ϕ' and ψ' are isomorphic on $\mathbf{S}$, and let $\pi(x_i) = \pi'(x_i)$ for $i = 1, \ldots, n$. Let f be a Boolean assignment of $x_1, \ldots, x_n$, and let g be a sorted assignment of $x_1, \ldots, x_n, y_1, y_2$ matching f. Without loss of generality, assume that $(g(y_1), g(y_2)) \notin \alpha_0$; for otherwise, pick two points c_1 and c_2 such that $(c_1, c_2) \notin \alpha_0$ (such points exist because $\alpha_0 < 1_{\mathbb{C}}$), settle $g(y_1) = c_1$ and $g(y_2) = c_1$, and note that g still matches f. First suppose that f satisfies ϕ. By Claim 4.2, g satisfies ϕ' on $\mathbf{S}$. By hypothesis, $g \circ \pi'$ satisfies ϕ' on $\mathbf{S}$. As $(g(y_1), g(y_2)) \notin \alpha_0$, and $f \circ \pi$ matches $g \circ \pi'$, we conclude by Claim 4.2 that $f \circ \pi$ satisfies ψ. Conversely, suppose that f does not satisfy ϕ. By Claim 4.2, g does not satisfy ϕ' on $\mathbf{S}$, and by hypothesis, $g \circ \pi'$ does not satisfy ψ' on $\mathbf{S}$; similarly, $f \circ \pi$ does not satisfy ψ. $\square$

This completes the first part of the proof. Second, we reduce from $\mathsf{PPISO}(\mathbf{S})$ to $\mathsf{PPISO}(\mathbf{A})$.

We import a technical lemma from [6]. Let X be a set, let $\{X_1, \ldots, X_k\}$ be a partition of X, and let π be a permuta-

tion of X. We say that π *fixes* X_i if $\{\pi(x) \mid x \in X_i\} = X_i$.

Lemma 4.5 *[6] Let σ be a signature, let $\mathbf{B}$ be a relational structure over σ, and let ϕ and ψ be pp-formulas on σ. For each $k \geq 2$, it is possible to construct in logspace, given a partition $\{X_1, \ldots, X_k\}$ of the set X of free variables of ϕ and ψ, two pp-formulas ϕ' and ψ' on σ satisfying: ϕ' and ψ' are isomorphic if and only if ϕ and ψ have an isomorphism that fixes $X_1, \ldots, X_k$.*

Let (ϕ, ψ) be a pair of sorted pp-formulas on $\mathbf{S}$ over variables $x_1, \ldots, x_n$ of sort B, and $y_1, \ldots, y_m$ of sort C, let (ϕ', ψ') be the pair of pp-formulas on $\mathbf{A}$ over variables $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m$ computed (in logspace) in part (i), and let (ϕ'', ψ'') be the pair of pp-formulas on $\mathbf{A}$ computed (in logspace) by Lemma 4.5 given the partition $\{\{x'_1, \ldots, x'_n\}, \{y'_1, \ldots, y'_m\}\}$. The following claim shows that the reduction is correct.

Say that a sectioning for a formula is *simple* if it has exactly as many sections as the free variables of the formula, and one type. Over any relational structure, two sorted pp-formulas are isomorphic with respect to the simple sectionings, if and only if they have an isomorphism; in fact, any bijection respects the simple sectionings.

Claim 4.6 *ϕ and ψ are isomorphic on $\mathbf{S}$ if and only if ϕ'' and ψ'' are isomorphic on $\mathbf{A}$ with respect to the simple sectionings.*

Proof. See Appendix C. □

This completes the second part of the proof. We conclude that $\mathsf{PPISO}(\mathbf{A})$ is MFISO-hard, and the proof of statement (ii) is complete. □

References

[1] A. Atserias. Conjunctive Query Evaluation by Search-Tree Revisited. *Theoretical Computer Science*, 371(3):155–168, 2007.

[2] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS'09*, pages 595–603, 2009.

[3] J. Berman, P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard. Varieties with Few Subalgebras of Powers. *Transactions of the American Mathematical Society*, 362(3):1445–1473, 2010.

[4] O. Beyersdorff, A. Meiera, M. Thomas, and H. Vollmer. The Complexity of Propositional Implication. *Information Processing Letters*, 109(18):1071–1077, 2009.

[5] V. Bodnarchuk, L. Kaluzhnin, V. Kotov, and B. Romov. Galois Theory for Post Algebras. I, II. *Cybernetics*, 5:243–252, 531–539, 1969.

[6] S. Bova, H. Chen, and M. Valeriote. On the Expression Complexity of Equivalence and Isomorphism of Primitive Positive Formulas. In A. A. Bulatov, M. Grohe, P. G. Kolaitis, and A. Krokhin, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, number 09441 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2010. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[7] A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of 18th IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 321–330, 2003.

[8] A. Bulatov. H-Coloring dichotomy revisited. *Theoretical Computer Science*, 349(1):31–39, 2005.

[9] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[10] A. A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM (JACM)*, 53, 2006.

[11] A. A. Bulatov and M. Valeriote. Recent results on the algebraic approach to the csp. In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 68–92. Springer, 2008.

[12] D. Geiger. Closed Systems of Functions and Predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.

[13] D. Hobby and R. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, 1988. Revised edition: 1996.

[14] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and Learnability Arising from Algebras with Few Subpowers. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2007.

[15] K. Kearnes, E. Kiss, and M. Valeriote. Minimal sets and varieties. *Transactions of the American Mathematical Society*, 350(1):1–41, 1998.

[16] P. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

[17] C. McGarry. k-Fold Systems of Projections and Congruence Modularity. M.Sc. Thesis, McMaster University, 2009.

[18] C. Papadimitriou and M. Yannakakis. On the Complexity of Database Queries. *Journal of Computer and System Sciences*, 58(3):407–427, 1999.

[19] S. Reith. On the Complexity of Some Equivalence Problems for Propositional Calculi. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2003.

[20] M. Vardi. The Complexity of Relational Query Languages. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1982.

Appendix

A Proof of Proposition 2.2

Proof. Let $\phi, \phi', \mathbf{B}, (s,t), (s',t')$ be an instance of PPISO with signature σ. Fix a $d \geq 1$ such that there is an injective mapping $r : B \to \{0,1\}^d$. Define $\hat{\sigma}$ to be the signature with the same relation symbols as σ, but where the arity of $R \in \hat{\sigma}$ is dk, where k is the arity of $R \in \sigma$. We create a boolean relational structure $\hat{\mathbf{B}}$ by defining $R^{\hat{\mathbf{B}}} = \{(r(b_1), \ldots, r(b_k)) \mid (b_1, \ldots, b_k) \in R^{\mathbf{B}}\}$ for all relation symbols R. The new instance is created as follows. The new pair of formulas (ψ, ψ') is created from the old pair (ϕ, ϕ') by replacing each variable v with a sequence of variables $v^1, \ldots, v^d$. The free variables of ψ and ψ' is thus $X' = \{x^i \mid x \in X, i \in [d]\}$ where X denotes the free variables of ϕ and ϕ'. The sectioning of ψ is given by $(\hat{s}, \hat{t}) : X' \to S \times (T \times [d])$, where $\hat{s}(x^i) = s(x)$ and $\hat{t}(x^i) = (t(x), i)$, and the sectioning of ψ' is given from that of ϕ' in an analogous way. It is straightforward to verify that this reduction is correct. □

B Proof of Theorem 2.6

Proof. We first treat powers; suppose $\mathbb{B} = \mathbb{A}^k$. Consider a problem $\mathsf{PPEQ}(\mathbf{B}) \in \mathsf{PPEQ}(\mathbb{A}_{\mathbf{B}})$, and let σ denote the signature of $\mathbf{B}$. Let σ' be the signature that has the same symbols as σ, but where the arity of a symbol of $R \in \sigma'$ is km, where m is the arity of $R \in \sigma$. Define $\mathbf{B}'$ to be the structure whose relation $R^{\mathbf{B}'}$ contains the tuple $(a_1^1, \ldots, a_1^k, \ldots, a_m^1, \ldots, a_m^k)$ if and only if the tuple $((a_1^1, \ldots, a_1^k), \ldots, (a_m^1, \ldots, a_m^k))$ belongs to the relation $R^{\mathbf{B}}$. Clearly, we have $\mathsf{PPEQ}(\mathbf{B}') \in \mathsf{PPEQ}(\mathbb{A})$. To reduce an instance (ϕ, ϕ') of $\mathsf{PPEQ}(\mathbf{B})$ to $\mathsf{PPEQ}(\mathbf{B}')$, we simply replace, in each of ϕ, ϕ', each variable v with a sequence of k variables $v^1, \ldots, v^k$. It is straightforward to verify that the original instance (ϕ, ϕ') was a yes instance if and only if the new formulas are. The same reduction applies to $\mathsf{PPCON}(\cdot)$. For the problem $\mathsf{PPISO}(\cdot)$, we apply the same transformation to the formulas, and define new sectionings as follows. Let $(s,t), (s',t') : X \to S \times T$ be the original sectionings. The new sectionings are denoted by $(s_1, t_1), (s_1', t_1') : X \to S \times T_1$ where $T_1 = T \times [k]$ and the mappings are defined by $s_1(v^i) = s(v)$, $s_1'(v^i) = s'(v)$, $t_1(v^i) = (t(v), i)$, and $t_1'(v^i) = (t(v), i)$. It is straightforward to verify that the original formulas are isomorphic with respect to $(s,t), (s',t')$ if and only if the new formulas are isomorphic with respect to $(s_1, t_1), (s_1', t_1')$.

In the case that $\mathbb{B}$ is a subalgebra or homomorphic image of $\mathbb{A}$, the result is proved in [6, Proposition 4] for $\mathsf{PPEQ}(\cdot)$, and from the argumentation there it is clear that exactly the same reduction works for $\mathsf{PPCON}(\cdot)$. For $\mathsf{PPISO}(\cdot)$, we apply the same reduction to the original pair of formulas (ϕ, ϕ') to obtain a new pair (ψ, ψ'), and associate the sectioning of ϕ with ψ, and that of ϕ' with ψ'. □

C Proof of Claim 4.6

Proof. ($\Rightarrow$) Let π be an isomorphism of ϕ and ψ, and let $\pi'(x_i') = x_j'$ if and only if $\pi(x_i) = x_j$ for $i, j \in \{1, \ldots, n\}$, and $\pi'(y_i') = y_j'$ if and only if $\pi(y_i) = y_j$ for $i, j \in \{1, \ldots, m\}$. We check that π' is an isomorphism of ϕ' and ψ'. Let g be an assignment of $x_1', \ldots, x_n', y_1', \ldots, y_m'$, and let f be the sorted assignment of $x_1, \ldots, x_n, y_1, \ldots, y_m$ matching g. Now, g satisfies ϕ' on $\mathbf{A}$ if and only if f satisfies ϕ on $\mathbf{S}$ (by Claim 4.3), if and only if $f \circ \pi$ satisfies ϕ on $\mathbf{S}$ (π is an isomorphism), if and only if $g \circ \pi'$ satisfies ϕ' on $\mathbf{A}$ (by Claim 4.3, since $f \circ \pi$ and $g \circ \pi'$ match). Therefore, π' is an isomorphism of ϕ' and ψ'. Moreover, π' fixes $\{x_1', \ldots, x_n'\}$ and $\{y_1', \ldots, y_m'\}$. Hence, by Lemma 4.5, ϕ'' and ψ'' are isomorphic on $\mathbf{A}$.

($\Leftarrow$) Let π'' be an isomorphism of ϕ'' and ψ'' on $\mathbf{A}$. By Lemma 4.5, there exists an isomorphism of ϕ' and ψ' on $\mathbf{A}$ that fixes $\{x_1', \ldots, x_n'\}$ and $\{y_1', \ldots, y_m'\}$. Let $\pi(x_i) = x_j$ if and only if $\pi'(x_i') = x_j'$ for $i, j \in \{1, \ldots, n\}$, and $\pi(y_i) = y_j$ if and only if $\pi'(y_i') = y_j'$ for $i, j \in \{1, \ldots, m\}$. Let f be a sorted assignment of $x_1, \ldots, x_n, y_1, \ldots, y_m$ and let g be an assignment of $x_1', \ldots, x_n', y_1', \ldots, y_m'$ matching f. It is easy to check that, by Claim 4.3, f satisfies ϕ on $\mathbf{S}$ if and only if $f \circ \pi$ satisfies ψ on $\mathbf{S}$. Hence, ϕ and ψ are isomorphic on $\mathbf{S}$. □