Journal of Machine Learning Research 20 (2019) 1-28 Submitted 12/17; Revised 1/19; Published 3/19

Learnability of Solutions to Conjunctive Queries

Hubie Chen H.CHEN@DCS.BBK.AC.UK
*Birkbeck, University of London,
London WC1E 7HX, United Kingdom*

Matthew Valeriote MATT@MATH.MCMASTER.CA
*Department of Mathematics & Statistics,
McMaster University,
Hamilton, Canada*

Editor: Mehryar Mohri

Abstract

The problem of learning the solution space of an unknown formula has been studied in multiple embodiments in computational learning theory. In this article, we study a family of such learning problems; this family contains, for each relational structure, the problem of learning the solution space of an unknown conjunctive query evaluated on the structure. A progression of results aimed to classify the learnability of each of the problems in this family, and thus far a culmination thereof was a positive learnability result generalizing all previous ones. This article completes the classification program towards which this progression of results strived, by presenting a negative learnability result that complements the mentioned positive learnability result. In addition, a further negative learnability result is exhibited, which indicates a dichotomy within the problems to which the first negative result applies. In order to obtain our negative results, we make use of universal-algebraic concepts.

Keywords: concept learning, computational learning theory, dichotomy theorems, reductions, universal algebra

1. Introduction

In this section we provide an overview of the question that we address in this paper and describe some related work as well as our contributions to its resolution.

1.1. Overview

The problem of learning the solution space of an unknown formula has long been of interest in computational learning theory. While the general problem of learning the solution space of even a propositional formula is known to be hard (Kearns and Valiant, 1994; Angluin and Kharitonov, 1995), researchers have considered many restricted versions of formula learning over the years, and have obtained a variety of learnability and non-learnability results (see for example Angluin, 1988; Angluin et al., 1992; Arias and Khardon, 2002; Bshouty et al., 2005; Jackson and Servedio, 2006; Bulatov et al., 2007; Idziak et al., 2010; Bshouty, 2013).

Conjunctive queries are formulas which are considered heavily in database theory and in the theory of constraint satisfaction. They can be defined logically as formulas built from predicate applications, equality of variables, conjunction, and existential quantification. For example, if $\mathbf{G} = (V, E)$ is an irreflexive graph (with vertex set V and edge set E, considered as a binary relation on V), then the conjunctive query

$$\exists y \exists z\, (E(x, y) \wedge E(y, z) \wedge E(z, x))$$

is true for a vertex $x \in V$ if and only if x is part of a 3-cycle of the graph.

The problem of deciding, given a conjunctive query and a *relational structure* (which defines the predicates of the query), whether or not the solution space of the query is non-empty, is a formulation of the *constraint satisfaction problem*, a very general NP-complete problem. One obtains a rich framework of problems, by considering, for each relational structure $\mathbf{B}$, the constraint satisfaction problem, denoted $\mathsf{CSP}(\mathbf{B})$, where the relational structure is fixed as $\mathbf{B}$; the computational aspects of this problem framework are of interest and have been explored in numerous contexts (see for example Creignou et al., 2001; Raghavendra, 2008; Allender et al., 2009; Chen, 2012; Bhattacharyya and Yoshida, 2013; Bulatov, 2013; Chen et al., 2016; Chen and Larose, 2017). Schaefer's celebrated dichotomy theorem (Schaefer, 1978) provides that, for each relational structure $\mathbf{B}$ with a two-element universe, $\mathsf{CSP}(\mathbf{B})$ is either polynomial-time decidable or is NP-complete. A line of research strives to obtain a complexity classification of the constraint satisfaction problem over all relational structures with finite universe; results here include sufficient conditions for tractability (Idziak et al., 2010; Barto and Kozik, 2014), a unifying explanation for known intractability proofs (Bulatov et al., 2005), and—as a culmination thus far—the classification of all such structures whose constraint satisfaction problem is polynomial-time tractable (Bulatov, 2017; Zhuk, 2017).

As a means of systematically exploring the boundary between learnability and non-learnability, an analogous framework has been considered in learning theory: for each relational structure $\mathbf{B}$, we may define a problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ wherein the aim is to learn the solution space of an unknown conjunctive query evaluated on $\mathbf{B}$ (refer to Section 2 for formal definitions). As three particular examples, consider the following. Other examples can be found in Barto et al. (2017).

Example 1 *Let $\mathbf{B}_{\mathrm{3SAT}}$ be the relational structure with domain $\{0, 1\}$ that has, for each triple $(a, b, c) \in \{0, 1\}^3$, the ternary relation $R_{(a,b,c)} = \{0, 1\}^3 \setminus \{(a, b, c)\}$. Then the set of solution spaces of the conjunctive queries on $\mathbf{B}_{\mathrm{3SAT}}$ is known to be the set of all Boolean relations.*

Example 2 *Let $\mathbf{B}_{\mathrm{HORN-3SAT}}$ be the relational structure with universe $\{0, 1\}$ whose relations consists of the four relations $\{0\}$, $\{1\}$, $R_{(1,1,0)}$ and $R_{(1,1,1)}$. It is readily verified that the solution spaces of conjunctive queries on $\mathbf{B}_{\mathrm{HORN-3SAT}}$ are exactly the solution spaces of conjunctions of propositional* Horn clauses*; these solution spaces can be equivalently characterized as those closed under the pointwise application of the Boolean AND ($\wedge$) operation (Creignou et al., 2001, Lemma 4.8).*

Example 3 *For a finite field $\mathbb{F} = (F; +, \cdot, -, 0, 1)$, let $\mathbf{V}_{\mathbb{F}}$ be the relational structure with universe F and whose relations are the singleton unary relations $\{f\}$, for $f \in F$; the graph of the function $x + y$; and, the graph of $\lambda_f(x) = f \cdot x$, for each $f \in F$. Then the solution spaces of conjunctive queries on $\mathbf{V}_{\mathbb{F}}$ are exactly the affine subspaces of the vector spaces $(\langle F, +, -, 0, \lambda_f \rangle_{f \in F})^n$, for $n \geq 1$.*

A primary research goal of this line of inquiry is to completely understand, over all finite structures $\mathbf{B}$, which problems of the form $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ are learnable and which are not.

1.2. Related Previous Work

Let us survey the main known results about the framework of learning problems $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.[1] Dalmau (1999) presented an analog of Schaefer's theorem, namely, a dichotomy theorem indicating, for each relational structure $\mathbf{B}$, which of the problems $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ are learnable. Precisely, this dichotomy theorem implies that each such problem is either polynomially learnable with equivalence queries, or is not polynomially predictable with membership queries. This dichotomy is sharp in that each problem that is polynomially learnable with equivalence queries is also polynomially predictable with membership queries (Littlestone, 1988, Section 4). The negative result, and all others under discussion, are proved under established cryptographic assumptions which are invoked in the present article (see Section 2.2), and the positive and negative results in the following discussion are proved in the two mentioned models, respectively.

Dalmau and Jeavons (2003) established a link between this framework and universal algebra; gave a general strategy for presenting positive results; and provided dichotomy theorems for two restricted classes of structures. Bulatov et al. (2007) gave a positive learnability result which applies to each relational structure having a so-called *generalized majority-minority polymorphism*. Later, Idziak et al. (2010) gave a positive learnability result generalizing all previous positive results; their result applies to any structure $\mathbf{B}$ for which all solution spaces have *small* (polynomial-size) generating sets, in a precise sense (see the discussion after Definition 9). They point out that all previous positive results were based on small generating sets, and hence that their result is a natural culmination of the progression of positive results.

1.3. Contributions

In this article, we complete the classification program towards which all of these previous works strive, by presenting a negative learnability result that complements the positive learnability result of Idziak et al. and hence that encompasses all previous negative learnability results in the framework at hand. We prove that for any structure $\mathbf{B}$ for which the small generating sets condition of Idziak et al. fails, it holds that $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is not polynomially predictable with membership queries. We accomplish this by reducing the problem of learning the solution space of an unknown propositional formula, denoted by $\mathcal{C}_{\mathrm{PF}}$, to the

[1] Let us mention that, in the existing literature, (in for example in Bulatov et al., 2007) some positive results are stated for queries where universal quantification is permitted in addition to predicate applications, equality of variables, conjunction, and existential quantification; let us refer to such queries as *quantified conjunctive queries*. Let $\mathbf{B}^*$ be the structure obtained from $\mathbf{B}$ by adding each element of the universe of $\mathbf{B}$ as a relation. These positive results typically apply to the structure $\mathbf{B}^*$ whenever they apply to a structure $\mathbf{B}$. For a structure of the form $\mathbf{B}^*$, it can be readily verified that the solution space of any quantified conjunctive query is also the solution space of a conjunctive query; this can be shown by transforming a quantified conjunctive query into a conjunctive query by using the additional relations to eliminate each instance of universal quantification.

As the main contribution of the present article is to present a negative result, we focus the present discussion on conjunctive queries.

problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$. We then refine this result by showing that if $\mathbf{B}$ fails to satisfy a weaker condition, that of having a Taylor polymorphism, then the problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is as hard to learn as a prediction problem concerning partial assignments to circuits.

In order to establish our negative results, we make significant use of universal-algebraic notions and results, which we now turn to elaborate on. Each structure $\mathbf{B}$ can be passed to an algebra, its so-called algebra of polymorphisms, and it is known that the complexity of learning $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is an invariant of this passage (that is, two structures that are passed to the same algebra have the same complexity of learning; see Proposition 4). We consider the variety generated by the algebra of a structure, which we show is justified (Proposition 5). If this variety is *congruence modular*, then we invoke a theorem, due to Barto (2018), which shows that the algebra of $\mathbf{B}$ has a property called *few subpowers*, and thus that the Idziak et al. positive result can be applied. (Barto's theorem resolved in the positive a conjecture known as the *Edinburgh Conjecture*, see Bova et al., 2013).

A main result of this article is the negative learnability result that, if the mentioned variety is *not* congruence modular, then the problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is hard to learn. In order to prove this, we make use of concepts developed in a previous work which also studied non-congruence modularity (Bova et al., 2013). In particular, we make use of a structural result established there (Lemma 21) which essentially shows that, to prove hardness, one can work with a relational structure which can be localized to behave as a set of *pentagons*, which are a certain type of relational structure. Exploiting this structural result in the context of learning, however, is far from obvious, and involves developing significantly more detailed reductions than those used in the previous work (Bova et al., 2013), which dealt with comparing the solution spaces of two given conjunctive queries. The reason the reductions need to be more detailed here is that, when reducing one problem to another, one needs to translate from one concept to a second in a way that closely preserves structure (in our case of studying learning of unknown formulas, reductions need to preserve structure of the solution spaces); this contrasts sharply with the earlier work (Bova et al., 2013), where reductions needed only preserve a single bit, namely, the answer to a decision problem. Indeed, as an intermediate step, we show the hardness of a natural term-learning problem on lattices, which may be of independent interest (Section 5).

To obtain our second negative result, we make use of another part of Bova et al. (2013) that deals with finite structures that fail to have a special type of polymorphism called a Taylor polymorphism. There it is shown that in the absence of a Taylor polymorphism, a finite structure $\mathbf{B}$ interprets, in a certain sense, a structure representing the 3-SAT problem. From this, we prove our hardness result.

2. Preliminaries

When P is a condition (such as a containment $x \in c$), we use $[P]$ to denote the value equal to 1 if P is true, and 0 if P is false. When $f : A \to B$ and $g : B \to C$ are functions, we sometimes use $g(f)$ to denote their composition.

2.1. Concept Learning

Our terminology and notation is based on those employed by Pitt and Warmuth (1990) and by Angluin and Kharitonov (1995).

We assume that objects are encoded over the binary alphabet $\{0, 1\}$, and use X to denote $\{0, 1\}^*$. When x is a string, we use $|x|$ to denote its length, and for each $n \in \mathbb{N}$, we use $X^{[n]}$ to denote $\{x \in X \ : \ |x| \leq n\}$. A *prediction problem* $\mathcal{C}$ is a subset of $X \times X$; when $(u, x) \in \mathcal{C}$, we refer to u as a *concept name* or *concept representation* (of $\mathcal{C}$). Relative to a prediction problem $\mathcal{C}$, the *concept represented by* u is defined as $\kappa_{\mathcal{C}}(u) = \{x \mid (u, x) \in \mathcal{C}\}$.

A *pwm-algorithm* (short for *prediction with membership queries algorithm*) is an algorithm A with the following properties. The algorithm A takes as input a bound $s \in \mathbb{N}$ on the size of the target concept representation, a bound $n \in \mathbb{N}$ on the length of examples, and an *accuracy bound* ϵ, a positive rational number. It may make three types of oracle calls, the responses to which are determined by an unknown target concept c and an unknown distribution D on $X^{[n]}$: (1) A *membership query* takes a string $x \in X$ as input and returns $[x \in c]$; (2) A request for a random classified example takes no input and returns a pair (x, b), where x is a string chosen independently according to D, and $b = [x \in c]$; (3) A request for an element to predict takes no input and returns a string x chosen independently according to D. The algorithm A may make any number of oracle calls of types 1 and 2; however, in any run, it must make exactly one oracle call of type 3 and then eventually halt with an output of 1 or 0 without making any further oracle calls.

A pwm-algorithm is said to run in polynomial time if its running time is bounded by a polynomial in s, n, and $1/\epsilon$. A pwm-algorithm A is said to *successfully* predict a prediction problem $\mathcal{C}$ if for each input (s, n, ϵ), each concept name $u \in X^{[s]}$ of $\mathcal{C}$, and for each probability distribution D on $X^{[n]}$, when A is run on (s, n, ϵ) and the oracle calls of type 1 and 2 are answered according to $c = \kappa_{\mathcal{C}}(u)$ and D, the probability that the output of A is not equal to $[x \in c]$ is bounded above by ϵ. A prediction problem is *polynomially predictable with membership queries* if there exists a pwm-algorithm that runs in polynomial time and successfully predicts $\mathcal{C}$.

2.2. Problems

We introduce the problems that will be of concern.

A *relational signature* is a finite set of *relation symbols*; each relation symbol has an arity $k \geq 0$ associated with it. Note that we assume that all relational signatures under discussion are finite. A *relational structure* $\mathbf{B}$ over a relational signature σ consists of a finite set B called its *universe* and, for each relation symbol $R \in \sigma$, a relation $R^{\mathbf{B}} \subseteq B^k$, where k is the arity of R. We generally use the letters $\mathbf{A}$, $\mathbf{B}$, ... to denote relational structures, and the corresponding letters A, B, ... to denote their respective universes. Note that we assume that all relational structures under discussion are *finite* in that each has a finite universe; nonetheless, we sometimes state this explicitly for emphasis. A *conjunctive query* on a relational signature σ is a first-order formula built from predicate applications $R(v_1, \ldots, v_k)$ (where $R \in \sigma$ and $v_1, \ldots, v_k$ are variables, with k equal to the arity of R), equality of variables $v = v'$, conjunction, and existential quantification. When $\mathbf{B}$ is a relational structure and $Q \subseteq B^k$ is a relation, we say that Q is *cq-definable* over $\mathbf{B}$ if there exists a conjunctive query $\phi(v_1, \ldots, v_k)$ such that $(b_1, \ldots, b_k)$ satisfies ϕ on $\mathbf{B}$ if and only if $(b_1, \ldots, b_k) \in Q$.

The prediction problems that we study are as follows. There is a problem for each relational structure $\mathbf{B}$. Each conjunctive query $\phi(V)$ over the signature of $\mathbf{B}$ is a concept representation, and its concept is the set that contains an assignment $f : V \to B$ if it holds that $\mathbf{B}, f \models \phi$, that is, if it satisfies $\phi(V)$ over $\mathbf{B}$. Formally, for each relational structure $\mathbf{B}$, we define $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ to be the prediction problem

$$\{(\phi(V), f) \mid \phi \text{ is a conjunctive query and } f : V \to B \text{ is a mapping such that } \mathbf{B}, f \models \phi\}.$$

Our hardness results for prediction problems are based on the hardness of predicting propositional formulas and of predicting Boolean circuits. By a propositional formula, we understand a formula built from propositional variables and the basis consisting of AND ($\wedge$), OR ($\vee$), and NOT ($\neg$), where the fan-in of AND and OR is assumed to be two. We define $\mathcal{C}_{\mathrm{PF}}$ as the prediction problem containing those pairs (θ, f) where θ is a propositional formula, and f is a propositional assignment to the variables of θ that satisfies θ. (Note that the existence of a pwm-algorithm for $\mathcal{C}_{\mathrm{PF}}$ is readily verified to be insensitive to our assumption of fan-in two for AND and OR gates.) The following cryptographic evidence is known for the hardness of learning $\mathcal{C}_{\mathrm{PF}}$. Let us refer to the following three hypotheses, studied in Kearns and Valiant (1994), as the *Kearns-Valiant hypotheses*: testing quadratic residues is intractable; inverting RSA encryption is intractable; factoring Blum integers is intractable.

Theorem 1 *(Angluin and Kharitonov, 1995, Corollary 3) Under the assumption that one of the Kearns-Valiant hypotheses holds, the prediction problem $\mathcal{C}_{\mathrm{PF}}$ is not polynomially predictable with membership queries.*

By a *Boolean circuit*, we understand a circuit built from input variables and the standard basis consisting of AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). We define $\mathcal{C}_{\mathrm{CIRC}}$ as the prediction problem containing those pairs (T, f) where T is a Boolean circuit, and f is an assignment to the variables of T that satisfies T. We define $\mathcal{C}_{\exists\mathrm{CIRC}}$ as the prediction problem containing those pairs $((T, U), f)$ where T is a Boolean circuit, U is a subset of the variables on which T is defined, and f is an assignment to U that can be extended to an assignment that satisfies T.

3. Reducibility and Hardness

In this section, we describe the notion of reduction that will be used throughout the paper (Section 3.1); we demonstrate how certain standard algebraic constructions are relevant in our learning context, and also present notions of algebra to be used (Section 3.2); and, we provide a certain learning problem on propositional formulas that will be wieldy (Section 3.3).

3.1. Oracular pwm-reducibility

We define an extension of the notion of *pwm-reduction* due to Angluin and Kharitonov (1995); we refer to our notion of reduction as *oracular pwm-reduction.*

An *oracular pwm-reduction* from a prediction problem $\mathcal{C}$ to a second prediction problem $\mathcal{C}'$ is a triple (f, g, H) where f and g are mappings and H is an algorithm with the following properties:

1. There exists a polynomial q such that for each $s, n \in \mathbb{N}$ and for each $u \in X^{[s]}$, it holds that $g(s, n, u)$ is a string with $|g(s, n, u)| \leq q(s, n, |u|)$.

2. For each $s, n \in \mathbb{N}$, for each $u \in X^{[s]}$, and for each $x \in X^{[n]}$, it holds that $x' = f(s, n, x)$ is a string such that $x \in \kappa_{\mathcal{C}}(u)$ if and only if $x' \in \kappa_{\mathcal{C}'}(g(s, n, u))$. Also, there exists a polynomial t such that f is computable in time $t(s, n, |x|)$.

3. For each $s, n \in \mathbb{N}$, for each $u \in X^{[s]}$, and for each $x' \in X^{[n]}$, the algorithm H, on input (s, n, x'), may submit strings $x \in X$ as queries to an oracle, which responds $[x \in \kappa_{\mathcal{C}}(u)]$; the algorithm's output must be $[x' \in \kappa_{\mathcal{C}'}(g(s, n, u))]$. The algorithm H is required to run in polynomial time (in s, n, and $|x'|$).

Let us remark that the existence of a pwm-reduction between two prediction problems immediately implies the existence of an oracular pwm-reduction: pwm-reducibility can be viewed as the special case of oracular pwm-reducibility where the algorithm H can make at most one oracle query and, in the case that this query is made, the result must be the output of H.

Proposition 2 *Let $\mathcal{C}$ and $\mathcal{C}'$ be prediction problems. If there exists an oracular pwm-reduction from $\mathcal{C}$ to $\mathcal{C}'$ and it holds that $\mathcal{C}'$ is polynomially predictable with membership queries, then $\mathcal{C}$ is also polynomially predictable with membership queries.*

The proof of Proposition 2 is extremely similar to that of Angluin and Kharitonov (1995, Lemma 2), so we only give the idea of the proof. Let A' be a pwm-algorithm that witnesses that $\mathcal{C}'$ is polynomially predictable with membership queries. We describe a pwm-algorithm A that witnesses that $\mathcal{C}$ is polynomially predictable with membership queries, as follows. When A is run on input (s, n, ϵ), it computes $s' = q(s, n, s)$ and $n' = t(s, n, n)$. It then performs a simulation of A' on input (s', n', ϵ). Oracle calls made by the simulation of A' are answered by A as follows.

1. When A' makes a membership query on string $x' \in X$, the algorithm A runs $H(s, n, x')$ using its own membership queries to respond to the oracle calls of H, and then returns the result to A'.

2. When A' requests a random classified example, the algorithm A makes a request for a random classified example to obtain (x, b), and then returns the pair $(f(s, n, x), b)$ to A'.

3. When A' requests an element to predict, the algorithm A requests an element x, and returns the string $f(s, n, x)$ to A'.

When the simulated algorithm A' halts with an output b, the algorithm A halts with the output b.

For each input (s, n, ϵ), for each concept name $u \in X^{[s]}$ of $\mathcal{C}$, and for each probability distribution D on $X^{[n]}$, set $u' = g(s, n, u)$ and set D' to be the induced distribution $f(s, n, D)$ on $X^{[n']}$. When the algorithm A is invoked on (s, n, ϵ) with u and D, in its simulation of A', membership queries are answered according to the concept $\kappa_{\mathcal{C}'}(u')$ and random classified examples are generated according to D'. The assumption that A' predicts correctly within an error bound of ϵ can be verified to imply that A will predict within an error bound of ϵ. This concludes our description of the proof of Proposition 2.

The following property, which is straightforward to verify, will be used tacitly.[2]

Proposition 3 *Oracular pwm-reducibility is transitive.*

We leave, to future research, a study of whether and how oracular pwm-reducibility is more general than pwm-reducibility.

3.2. Algebras and Varieties

We make use of basic notions from universal algebra, and suggest Burris and Sankappanavar (1981) and McKenzie et al. (1987) as references. For our purposes in this article, an *algebra* is a pair $(A; F)$ consisting of a nonempty set A, the *universe* of the algebra, and a set F of finitary operations on A, the set of *basic operations* of the algebra. An algebra is *finite* if its universe is finite; we deal here mainly with finite algebras. A *term operation* (or sometimes just term) of an algebra $\mathbb{A}$ is any operation on A that can be obtained from the basic operations of $\mathbb{A}$ and the projection maps on A by composition. The *variety generated by an algebra* $\mathbb{A}$, denoted by $\mathcal{V}(\mathbb{A})$, is the smallest class of algebras containing $\mathbb{A}$ that is closed under taking homomorphic images, subalgebras, and products. An operation $f : B^m \to B$ is a *polymorphism* of a relation $Q \subseteq B^k$ if for any m tuples $(b_1^1, \dots, b_k^1), \dots, (b_1^m, \dots, b_k^m)$ in Q, the tuple $(f(b_1^1, \dots, b_1^m), \dots, f(b_k^1, \dots, b_k^m))$ is in Q. A relational structure $\mathbf{B}$ is *compatible* with an algebra having the same universe B if for each operation $f : B^m \to B$ of the algebra, it holds that f *is a polymorphism of* $\mathbf{B}$, by which is meant, f is a polymorphism of each relation of $\mathbf{B}$. We similarly speak of a single relation or a set of relations being *compatible* with an algebra. For a relational structure $\mathbf{B}$, we define $\mathbb{A}(\mathbf{B})$ to be the algebra with universe B and whose operations are the polymorphisms of $\mathbf{B}$.

We will make use of the following two facts. The first was established in previous work, and shows the relevance of the algebra of a structure to the problem framework at hand. The second shows the relevance of the variety of the algebra of a structure thereto.

Proposition 4 *(follows from Dalmau and Jeavons, 2003, Proof of Lemma 9) Suppose that* $\mathbf{B}$ *and* $\mathbf{B}'$ *are relational structures with the same universe and such that* $\mathbf{B}$ *is compatible with* $\mathbb{A}(\mathbf{B}')$. *Then there exists an oracular pwm-reduction from* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}')$.

Proposition 5 *Suppose that* $\mathbb{B}$ *is a finite algebra, and that* $\mathbf{A}$ *is a finite structure which is compatible with an algebra in* $\mathcal{V}(\mathbb{B})$. *Then, there exists a relational structure* $\mathbf{B}$ *which is compatible with* $\mathbb{B}$ *such that there exists an oracular pwm-reduction from* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.

It is well-known that a finite algebra is in $\mathcal{V}(\mathbb{B})$ if and only if it is a homomorphic image of a subalgebra of a finite power of $\mathbb{B}$. Proposition 5 thus follows immediately from the three following lemmas.

[2] We remark that, strictly speaking, transitivity of oracular pwm-reducibility is not needed to derive the main result of the paper. Our main result shows that for certain relational structures $\mathbf{B}$, the prediction problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is not polynomially predictable with membership queries unless $\mathcal{C}_{\mathrm{PF}}$ is as well. To establish this, it suffices to give a sequence of pwm-reductions from $\mathcal{C}_{\mathrm{PF}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ (which is what we do) and then invoke Proposition 2.

Lemma 6 *Suppose that $\mathbb{B}$ is a finite algebra and that $\mathbf{A}$ is a finite structure which is compatible with a finite power $\mathbb{A} = \mathbb{B}^n$ of $\mathbb{B}$. Then there exists a relational structure $\mathbf{B}$ compatible with $\mathbb{B}$ such that there exists an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.*

Proof Let $\mathbf{A}$ be a structure on signature $\sigma_{\mathbf{A}}$ which is compatible with $\mathbb{A}$. Define $\sigma_{\mathbf{B}}$ to be the signature having the same symbols as $\sigma_{\mathbf{A}}$, but where the arity of a symbol $R \in \sigma_{\mathbf{B}}$ is nk, where k is the arity of R in $\sigma_{\mathbf{A}}$. Define $\mathbf{B}$ so that a relation $R^{\mathbf{B}}$ contains a tuple $(b_1^1, \ldots, b_1^n, \ldots, b_k^1, \ldots, b_k^n)$ if and only if $R^{\mathbf{A}}$ contains $((b_1^1, \ldots, b_1^n), \ldots, (b_k^1, \ldots, b_k^n))$.

The reduction (f, g, H) is as follows. For a conjunctive query $\phi_{\mathbf{A}}$ on $\sigma_{\mathbf{A}}$, the function g is defined so that $g(s, m, \phi_{\mathbf{A}})$ is equal to $\phi_{\mathbf{B}}$, where $\phi_{\mathbf{B}}$ is derived from $\phi_{\mathbf{A}}$ by replacing each variable v by a tuple $(v^1, \ldots, v^n)$ of variables. The mapping f is defined so that, when h is an assignment from V to $A = B^n$, $f(s, m, h)$ is the map $h' : \{v^1, \ldots, v^n \mid v \in V\} \to B$ such that the following condition holds: for each $v \in V$, it holds that $h(v) = (h'(v^1), \ldots, h'(v^n))$. The algorithm $H(s, m, h' : \{v^1, \ldots, v^n \mid v \in V\} \to B)$ calculates the mapping $h : V \to A$ defined according to the just-stated condition, submits h to its oracle, and outputs the result. This reduction is correct, as for a pair of assignments h, h' satisfying the condition, it holds that h satisfies $\phi_{\mathbf{A}}$ on $\mathbf{A}$ if and only if h' satisfies $\phi_{\mathbf{B}}$ on $\mathbf{B}$. ■

Lemma 7 *Suppose that $\mathbb{B}$ is a finite algebra and that $\mathbf{A}$ is a finite structure which is compatible with a subalgebra $\mathbb{A}$ of $\mathbb{B}$. Then there exists a relational structure $\mathbf{B}$ compatible with $\mathbb{B}$ such that there exists an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.*

Proof Suppose that $\mathbf{A}$ is a structure on signature $\sigma_{\mathbf{A}}$ compatible with $\mathbb{A}$. Define $\sigma_{\mathbf{B}}$ to be a signature equal to $\sigma_{\mathbf{A}}$ but expanded by a relation symbol U of arity 1. Let $\mathbf{B}$ be the structure over $\sigma_{\mathbf{B}}$ with universe B where $R^{\mathbf{B}} = R^{\mathbf{A}}$ for each $R \in \sigma_{\mathbf{A}}$ and where $U^{\mathbf{B}} = A$.

The reduction (f, g, H) is as follows. For a conjunctive query $\phi_{\mathbf{A}}$ on $\sigma_{\mathbf{A}}$, define $g(s, n, \phi_{\mathbf{A}})$ to be the formula $\phi_{\mathbf{B}}$ which is obtained from $\phi_{\mathbf{A}}(V)$ by replacing each predicate application $R(v_1, \ldots, v_k)$ by $R(v_1, \ldots, v_k) \wedge U(v_1) \wedge \cdots \wedge U(v_k)$ to obtain $\psi(V)$, and then defining $\phi_{\mathbf{B}} = \psi(V) \wedge \bigwedge_{v \in V} U(v)$. Essentially, $\phi_{\mathbf{B}}$ is obtained from $\phi_{\mathbf{A}}$ by restricting all free and used variables to take on values in $U^{\mathbf{B}} = A$. It is straightforward to verify that $\phi_{\mathbf{A}}$ and $\phi_{\mathbf{B}}$ have the same solutions (with respect to the structures $\mathbf{A}$ and $\mathbf{B}$, respectively). Hence, f may be defined by $f(s, n, h) = h$, and $H(s, n, h')$ may be defined as the algorithm that passes h' to its oracle and outputs the result. ■

A *congruence* of an algebra $\mathbb{A} = (A; F)$ is an equivalence relation on A that is compatible with $\mathbb{A}$. Suppose that θ is a congruence of $\mathbb{A}$. We use a^θ to denote the equivalence class of θ containing $a \in A$. For each operation $f \in F$, the operation f^θ defined by $f^\theta(a_1^\theta, \ldots, a_k^\theta) = (f(a_1, \ldots, a_k))^\theta$ is well-defined. An algebra is a *homomorphic image* of $\mathbb{A}$ if it is isomorphic to an algebra of the form $(A^\theta; F^\theta)$ where $A^\theta = \{a^\theta \mid a \in A\}$ and $F^\theta = \{f^\theta \mid f \in F\}$.

Lemma 8 *Suppose that $\mathbb{B}$ is a finite algebra and that $\mathbf{A}$ is a finite structure which is compatible with a homomorphic image $\mathbb{A}$ of $\mathbb{B}$. Then there exists a relational structure $\mathbf{B}$ compatible with $\mathbb{B}$ such that there exists an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.*

Proof We assume that $\mathbb{A}$ is equal to (A^θ, F^θ) where θ is a congruence of $\mathbb{B}$. Let $\mathbf{A}$ be a structure over signature $\sigma_\mathbf{A}$ which is compatible with $\mathbb{A}$. Define $\mathbf{B}$ to be the structure over signature $\sigma_\mathbf{A}$ defined by $R^\mathbf{B} = \{(b_1, \dots, b_k) \mid (b_1^\theta, \dots, b_k^\theta) \in R^\mathbf{A}\}$. For any conjunctive query $\phi(V)$ over $\sigma_\mathbf{A}$, it is straightforward to verify that an assignment $h : V \to A$ satisfies ϕ on $\mathbf{A}$ if and only if one (equivalently, every) assignment $h' : V \to B$ with $(h'(v))^\theta = h(v)$ satisfies ϕ on $\mathbf{B}$.

The following is thus a reduction. Define $g(s, n, \phi) = \phi$, define $f(s, n, h)$ to be an assignment h' defined as above, and define $H(s, n, h')$ to be the algorithm that passes the assignment h defined by $h(v) = (h'(v))^\theta$ to its oracle and returns the result. ■

A *lattice* is an algebra $(L; \wedge, \vee)$ where each of the operations $\wedge$ and $\vee$ is binary, idempotent, commutative, and associative; and, the absorption law $a \wedge (a \vee b) = a \vee (a \wedge b) = a$ holds. A lattice naturally induces a partial order $\leq$ defined by $a \leq b$ if and only if $a \wedge b = a$. A lattice is *distributive* if it satisfies the identity $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. We say that a lattice is *non-trivial* if its universe has size strictly greater than 1. By a *lattice term*, we refer to a term built from variables and the two operation symbols $\wedge$ and $\vee$. The congruences of an algebra naturally form a lattice. An algebra $\mathbb{A}$ is *congruence modular* if its lattice of congruences satisfies the modular law: $x \leq y \to x \vee (y \wedge z) = y \wedge (x \vee z)$. A class of algebras is *congruence modular* if each algebra therein is congruence modular.

Varieties whose members are all congruence modular have special properties and have been extensively studied in the universal algebra literature. For example, there is a useful notion of a commutator for such varieties and it has been used to establish strong structural results (Freese and McKenzie, 1987). Congruence modular varieties can be characterized in terms of the existence of special terms, called *Day terms* or *Gumm terms*, that satisfy particular equations (see Freese and Valeriote, 2009, Section 8). In the special case where the variety is of the form $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ for some finite relational structure $\mathbf{B}$, Barto (2018) has established a deep result that shows that this variety will be congruence modular if and only if $\mathbf{B}$ has a special polymorphism called an *edge term*.

Definition 9 *Let A be a nonempty set and $k > 1$. A k*-edge operation *on A is a $k+1$-ary operation $t(x_1, x_2, \dots, x_{k+1})$ that satisfies the following equations:*

$$\begin{aligned}
t(y, y, x, x, x, \dots, x) &= x\\
t(y, x, y, x, x, \dots, x) &= x\\
t(x, x, x, y, x, \dots, x) &= x\\
t(x, x, x, x, y, \dots, x) &= x\\
&\vdots\\
t(x, x, x, x, x, \dots, y) &= x.
\end{aligned}$$

A structure $\mathbf{B}$ has an edge term polymorphism *if for some $k > 1$ it has a polymorphism that is a k-edge operation.*

Example 4 *1. If $\mathbb{G} = (G, \cdot, {}^{-1}, e)$ is a group then the term operation $t(x_1, x_2, x_3) = x_2 \cdot x_1^{-1} \cdot x_3$ is a 2-edge operation on G.*

2. *If* $\mathbf{L} = (L; \wedge, \vee)$ *is a lattice, then the term operation*

$$m(x_1, x_2, x_3, x_4) = (x_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (x_3 \wedge x_4)$$

is a 3-edge operation on L.

The notion of an edge term was introduced and investigated in Berman et al. (2010). It is shown in that paper that for a finite algebra $\mathbb{A}$, having a k-edge term for some $k > 1$ is equivalent to the *few subpowers property* for $\mathbb{A}$: there is some polynomial $p(n)$ such that for each $n > 0$, the number of subalgebras of $\mathbb{A}^n$ is less than $2^{p(n)}$. In general, the number of subalgebras of $\mathbb{A}^n$ can grow doubly exponentially in n, so this is a special property of an algebra. It is also shown that having a k-edge term for some $k > 1$ is equivalent to the subalgebras of finite powers of $\mathbb{A}$ having "small generating sets", which means that there is some polynomial $g(n)$ such that for $n > 0$, each subalgebra of $\mathbb{A}^n$ has a generating set of size at most $g(n)$.

In the companion paper (Idziak et al., 2010) it is shown that if a finite relational structure $\mathbf{B}$ has an edge term polymorphism then the prediction problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is polynomially exactly learnable with improper equivalence queries, using a concept representation that is polynomially evaluable. The key property of $\mathbf{B}$ used in the proof of this fact is that the solution space of a conjunctive query over $\mathbf{B}$ is a subalgebra of a power of the algebra $\mathbb{A}(\mathbf{B})$ and so has a small generating set. This allows for a nice compact representation of each concept that is represented by a conjunctive query over $\mathbf{B}$.

A weaker example of a term condition, along the lines of having an edge term, is that of having a *Taylor term* or *Taylor operation*. A *Taylor operation* on a set A is an n-ary function $t : A^n \to A$ that is *idempotent* in that it satisfies the equation $t(x, x, \ldots, x) = x$ and, for each $1 \leq i \leq n$, an equation of the form $t(v_1, v_2, \ldots, v_n) = t(w_1, w_2, \ldots, w_n)$, where the v_j and w_j are variables from the set $\{x, y\}$ and $v_i \neq w_i$. A Taylor term of an algebra $\mathbb{A}$ is a term operation of $\mathbb{A}$ that is a Taylor operation. It can be readily verified that any k-edge operation on a set is also a Taylor operation on it. In Taylor (1977) it is shown that, by some measure, the condition of having a Taylor term is the weakest sort of term condition of this type that an algebra (or variety) can satisfy.

A *Taylor polymorphism* of a structure $\mathbf{B}$ is a polymorphism of $\mathbf{B}$ that is also a Taylor operation. The only polymorphisms of the structure $\mathbf{B}_{3\mathrm{SAT}}$ from Example 1 are the projection maps and so $\mathbf{B}_{3\mathrm{SAT}}$ does not have a Taylor polymorphism (Barto et al., 2017). On the other hand, the structure $\mathbf{B}_{\mathrm{HORN-3SAT}}$ from Example 2 has $x \wedge y$ as a Taylor polymorphism and $\mathbf{V}_{\mathbb{F}}$ from Example 3 has $x - y + z$ (this polymorphism ensures that $\mathcal{V}(\mathbb{A}(\mathbf{V}_{\mathbb{F}}))$ is congruence modular).

Taylor operations have played a central role in the investigation of the constraint satisfaction problem over a finite relational structure or finite algebra. The ultimate resolution of the Feder-Vardi Dichotomy Conjecture (Feder and Vardi, 1999) can be expressed in terms of Taylor polymorphisms, but also in terms of related conditions that are equivalent for finite structures and algebras. In Maróti and McKenzie (2008) it is shown that a finite algebra $\mathbb{A}$ will have a Taylor term if and only if it has a term that is a weak near-unanimity operation. A *weak near-unanimity operation* on a set A is an operation $w(x_1, x_2, \ldots, x_k)$ for some $k > 1$ that satisfies the equations $w(x, x, \ldots, x) = x$ and

$$w(y, x, \ldots, x) = w(x, y, \ldots, x) = \cdots = w(x, x, \ldots, x, y).$$

It is not hard to see that any weak near-unanimity operation is a Taylor operation but quite challenging to show that if a finite algebra has a Taylor term, then it also has a weak near-unanimity term operation. A similar result holds for finite structures: $\mathbf{B}$ will have a Taylor polymorphism if and only if it has a weak near-unanimity polymorphism.

Bulatov (2017) and Zhuk (2017) have shown that for a finite relational structure $\mathbf{B}$ such that for each $b \in B$, the unary relation $\{b\}$ is a relation of $\mathbf{B}$, $\mathsf{CSP}(\mathbf{B})$ is polynomial-time decidable if $\mathbf{B}$ has a Taylor polymorphism (or equivalently, a weak near-unanimity polymorphism). It has been known for some time (Jeavons, 1998) that in the absence of a Taylor polymorphism, this problem is NP-complete and so combining these results (and some other basic facts) leads to the following dichotomy: for a finite relational structure $\mathbf{B}$, $\mathsf{CSP}(\mathbf{B})$ is polynomial-time decidable or is NP-complete.

3.3. Propositional Formulas

By log, we indicate the logarithm base 2. When θ is a formula or a term, we define $\text{depth}(\theta)$ to be the maximum length of a path from the root of θ (viewed as a tree) to a leaf; we define $\text{leafsize}(\theta)$ to be the number of leaves of θ (again, viewed as a tree). Define $\mathcal{C}_{\text{log-MPF}}$ to be the subset of $\mathcal{C}_{\text{PF}}$ that contains a pair $(\theta, h) \in \mathcal{C}_{\text{PF}}$ when θ is *monotone* (that is, does not contain any instance of negation $(\neg)$) and when $\text{depth}(\theta) \leq 6 + 6\log(\text{leafsize}(\theta))$.

The following proposition is readily derivable using Spira's lemma and known techniques for representing a propositional formula as a monotone propositional formula.

Proposition 10 *There exists an oracular pwm-reduction from $\mathcal{C}_{\text{PF}}$ to $\mathcal{C}_{\text{log-MPF}}$.*

Let us prove this proposition, in two steps. We first observe a reduction to the following intermediate problem. Define $\mathcal{C}_{\text{log-PF}}$ to be the subset of $\mathcal{C}_{\text{PF}}$ that contains a pair $(\theta, h) \in \mathcal{C}_{\text{PF}}$ when θ has $\text{depth}(\theta) \leq 1 + 4\log(\text{leafsize}(\theta))$.

Lemma 11 *(derivable from Spira's Lemma; see the presentation/discussion in Bonet and Buss, 1994)*[3] *Let ϕ be a propositional formula; then there exists an equivalent propositional formula ϕ' such that* $\text{depth}(\phi') \leq 1 + 4\log(\text{leafsize}(\phi))$ *and such that* $\text{leafsize}(\phi) \leq \text{leafsize}(\phi') \leq \text{leafsize}(\phi)^3$*. It thus holds that* $\text{depth}(\phi') \leq 1 + 4\log(\text{leafsize}(\phi'))$.

The following proposition is readily derived from Lemma 11.

Proposition 12 *There exists an oracular pwm-reduction from $\mathcal{C}_{\text{PF}}$ to $\mathcal{C}_{\text{log-PF}}$.*

It then remains to give a reduction from $\mathcal{C}_{\text{log-PF}}$ to $\mathcal{C}_{\text{log-MPF}}$, which is what we now do.

Proposition 13 *There exists an oracular pwm-reduction from $\mathcal{C}_{\text{log-PF}}$ to $\mathcal{C}_{\text{log-MPF}}$.*

[3] We remark that to guarantee $\text{leafsize}(\phi) \leq \text{leafsize}(\phi')$, one can repeatedly apply the transformation described in Bonet and Buss (1994) but leaving in the constants (0 and 1) that are introduced. At the end, one can then replace the constants 0 and 1 by $(v \wedge \neg v)$ and $(v \vee \neg v)$, respectively, where v is some variable.

Proof The proof proceeds along the lines of that of Dalmau (1999, Lemma 33). We describe the parts of the reduction. When θ is a propositional formula on variables $v_1, \ldots, v_m$, the mapping g is defined so that $\psi = g(s, n, \theta)$ is logically equivalent to

$$(v_1 \vee v_1') \wedge \cdots \wedge (v_m \vee v_m') \wedge [(v_1 \wedge v_1') \vee \cdots \vee (v_m \wedge v_m') \vee \theta].$$

Note that this expression is written using a conjunction of high fan-in and a disjunction of high fan-in. In each case, each can be rewritten as a formula where conjunction and disjunction have fan-in 2; then, the high fan-in conjunction and high fan-in disjunction are replaced with formulas having depth at most $1 + \log(m + 1)$. By rewriting in this fashion, we obtain ψ. As the depth of each disjunction $(v_i \vee v_i')$ has depth 1, we can naively bound the depth of ψ by $2 * (1 + \log(\text{leafsize}(\psi))) + 1 + \text{depth}(\theta)$ which is upper bounded by $6 + 6\log(\text{leafsize}(\psi))$. The function induced by the formula $g(s, n, \theta)$ evaluates to θ if for each i it holds that $v_i \neq v_i'$; to 0 if there exists an i such that $v_i = v_i'$; and, to 1 otherwise. For each assignment h from a set of variables $\{v_1, \ldots, v_m\}$ to $\{0, 1\}$, the mapping f is defined as $f(s, n, h) = h'$ where h' is the unique extension of h such that $h'(v_i') = \neg v_i$ for each i. The algorithm H, on input $(s, n, h' : \{v_1, v_1', \ldots, v_m, v_m'\} \to \{0, 1\})$, outputs the result of a query call on the restriction of h' to $\{v_1, \ldots, v_m\}$, if for each i it holds that $h'(v_i) \neq h'(v_i')$; 0 if there exists an i such that $h'(v_i) = h'(v_i')$; and, 1 otherwise. ■

4. Main Theorems

We are now in a position to present the main theorems and to explain how they will follow from the results in the following sections.

Theorem 14 *Let* $\mathbf{B}$ *be a finite relational structure.*

- *If the variety* $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ *is congruence modular, then the prediction problem* $\mathcal{C}_{\text{CQ}}(\mathbf{B})$ *is polynomially exactly learnable with improper equivalence queries, using a concept representation that is polynomially evaluable.*
- *Otherwise, there is an oracular pwm-reduction from* $\mathcal{C}_{\text{PF}}$ *to* $\mathcal{C}_{\text{CQ}}(\mathbf{B})$ *and so the prediction problem* $\mathcal{C}_{\text{CQ}}(\mathbf{B})$ *is not polynomially predictable with membership queries unless* $\mathcal{C}_{\text{PF}}$ *is as well, and hence (by Theorem 1) not unless each of the Kearns-Valiant hypotheses fails.*

Let us remark that the following is known: each problem that is polynomially exactly learnable with improper equivalence queries under a polynomially evaluable concept representation is polynomially predictable with membership queries (see for example Angluin, 1988, Section 2.4).

Proof If the variety $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular, then by Barto's theorem (Barto, 2018), it holds that this variety has *few subpowers* and that there is a k-edge polymorphism of $\mathbf{B}$; thus, the Idziak et al. result (Idziak et al., 2010, Corollary 5.6) applies. For more details, see the discussion after Definition 9 above. If this variety is not congruence modular, then Proposition 10, Theorem 18, Theorem 20, and Theorem 22 yield a sequence of oracular pwm-reductions from the prediction problem $\mathcal{C}_{\mathrm{PF}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$, where $\mathbf{A}$ is a structure compatible with an algebra in the variety; an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ exists by appeal to Propositions 5 and 4. Hence, $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is not polynomially predictable with membership queries unless $\mathcal{C}_{\mathrm{PF}}$ is as well, by Proposition 2. ■

The following theorem refines the understanding of the problems that are hard according to Theorem 14.

Theorem 15 *Let* $\mathbf{B}$ *be a finite relational structure.*

- *If* $\mathbf{B}$ *has a Taylor polymorphism, then there is an oracular pwm-reduction from* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ *to* $\mathcal{C}_{\mathrm{CIRC}}$*.*
- *Otherwise, there is an oracular pwm-reduction from* $\mathcal{C}_{\exists\mathrm{CIRC}}$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$*.*

We note that if $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular, then $\mathbf{B}$ will have a Taylor polymorphism and so the positive result of Theorem 15 applies to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$; however, the positive result of Theorem 14 is stronger than that of Theorem 15, since $\mathcal{C}_{\mathrm{CIRC}}$ is not known to be learnable in the sense of the first item of Theorem 14. Hence, Theorem 15 is in essence a theorem concerning the problems that are hard according to Theorem 14.

We believe that it is natural to expect that Theorem 15 gives a true dichotomy in that the problem $\mathcal{C}_{\exists\mathrm{CIRC}}$ is strictly harder than the problem $\mathcal{C}_{\mathrm{CIRC}}$. A heuristic reason for believing this is that the computational decision problem of deciding if a given pair (T, f) belongs to $\mathcal{C}_{\mathrm{CIRC}}$ is polynomial-time computable (instantiate the input variables according to f, and then evaluate the circuit), whereas the computational decision problem of deciding if a given pair $((T, U), f)$ belongs to $\mathcal{C}_{\exists\mathrm{CIRC}}$ is NP-complete (in the case that $U = \emptyset$, this problem amounts to circuit satisfiability).

Before starting the proof of Theorem 15, recall that for a relational structure $\mathbf{B}$, $\mathsf{CSP}(\mathbf{B})$ is the computational problem of deciding, given a conjunctive query, whether or not its solution space over $\mathbf{B}$ is nonempty. Equivalently, $\mathsf{CSP}(\mathbf{B})$ is the problem of deciding, given a conjunctive query ψ whose variables are all existentially quantified, whether or not $\mathbf{B} \models \psi$, that is, whether or not ψ evaluates to true on $\mathbf{B}$. Define $\mathbf{B}^*$ to be the structure derived from $\mathbf{B}$ by adding, for each element $b \in B$, a relation $U_b^{\mathbf{B}^*} = \{b\}$.

Proof If $\mathbf{B}$ has a Taylor polymorphism, we argue as follows. The structure $\mathbf{B}^*$ also has a Taylor polymorphism, since a Taylor operation is idempotent. We obtain from Bulatov (2017) or Zhuk (2017) that $\mathsf{CSP}(\mathbf{B}^*)$ is polynomial-time decidable; hence, there is a polynomial-time computable function that, given a conjunctive query $\phi(V)$, outputs a Boolean circuit T such that for each assignment $f : V \to B$, it holds that $\mathbf{B}, f \models \phi$ if and only if f satisfies T. The existence of the claimed oracular pwm-reduction follows immediately.

If $\mathbf{B}$ does not have a Taylor polymorphism then by Theorem 23 there will be an oracular pwm-reduction from $\mathcal{C}_{\exists\mathrm{CIRC}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$. ■

From Theorems 14 and 15, we obtain the following corollary—a trichotomy theorem.

Corollary 16 *Let* $\mathbf{B}$ *be a finite relational structure.*

- *If the variety $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular, then the prediction problem $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ is polynomially exactly learnable with improper equivalence queries, using a concept representation that is polynomially evaluable.*
- *If the variety $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is not congruence modular but $\mathbf{B}$ has a Taylor polymorphism, then there is an oracular pwm-reduction from $\mathcal{C}_{\mathrm{PF}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$, as well as an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ to $\mathcal{C}_{\mathrm{CIRC}}$.*
- *Otherwise, there is an oracular pwm-reduction from $\mathcal{C}_{\exists\mathrm{CIRC}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.*

We can apply this result to the examples presented in the introduction to determine the complexity of learning the solution spaces of conjunctive queries over them. Since the structure $\mathbf{B}_{3\mathrm{SAT}}$ has no non-trivial polymorphisms, then it has no Taylor polymorphism and so there is an oracular pwm-reduction from $\mathcal{C}_{\exists\mathrm{CIRC}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{3\mathrm{SAT}})$. The structure $\mathbf{B}_{\mathrm{HORN-3SAT}}$ has a Taylor polymorphism (the operation $x \wedge y$) but $\mathcal{V}(\mathbb{A}(\mathbf{B}_{\mathrm{HORN-3SAT}}))$ is not congruence modular since this variety is essentially the variety of semilattices. So, there is an oracular pwm-reduction from $\mathcal{C}_{\mathrm{PF}}$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{\mathrm{HORN-3SAT}})$, as well as an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{\mathrm{HORN-3SAT}})$ to $\mathcal{C}_{\mathrm{CIRC}}$. Since $\mathcal{V}(\mathbb{A}(\mathbf{V}_{\mathbb{F}}))$ is congruence modular for any finite field $\mathbb{F}$, then $\mathcal{C}_{\mathrm{CQ}}(\mathbf{V}_{\mathbb{F}})$ is polynomially exactly learnable with improper equivalence queries, using a concept representation that is polynomially evaluable.

It is worth comparing the above trichotomy theorem with the CSP dichotomy theorem (Bulatov, 2017; Zhuk, 2017). The comparison works well up to a point, namely, if $\mathbf{B}$ falls into either of the first two cases of Corollary 16 then it will have a Taylor polymorphism, and so $\mathsf{CSP}(\mathbf{B})$ will be polynomial-time decidable. On the other hand, for any relational structure $\mathbf{B}$ that has a constant function as a polymorphism, $\mathsf{CSP}(\mathbf{B})$ is trivially polynomial-time decidable, even if $\mathbf{B}$ fails to have a Taylor polymorphism. So there are structures $\mathbf{B}$ such that $\mathsf{CSP}(\mathbf{B})$ is polynomial-time decidable while $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$ admits an oracular pwm-reduction from $\mathcal{C}_{\exists\mathrm{CIRC}}$. It is the case that $\mathbf{B}$ will fall into the third case of Corollary 16 if and only if $\mathsf{CSP}(\mathbf{B}^*)$ is NP-complete.

Let us now present a theorem that addresses the effectivity of the dichotomy conditions of Theorems 14 and 15. That is, we address the complexity of deciding, given a relational structure $\mathbf{B}$, whether $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular, and whether $\mathbf{B}$ has a Taylor polymorphism.

Theorem 17 *Let $\mathbf{B}$ be a finite relational structure.*

1. *There is an EXPTIME algorithm that decides if the variety $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular.*
2. *The problem of deciding if $\mathbf{B}$ has a Taylor polymorphism is in NP.*

Proof Part 1 follows from the characterization of congruence modular varieties given by Day or Gumm (consult Section 8 of Freese and Valeriote, 2009). Using Gumm's characterization, to determine if $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ is congruence modular one need only search amongst the ternary functions on B for a finite sequence of polymorphisms of $\mathbf{B}$ that satisfy a specified set of equations. This search can be carried out by an algorithm whose running time is bounded by an exponential function in the size of $\mathbf{B}$. A full discussion of the relevant details can be found in Section 8 of Freese and Valeriote (2009).

For Part 2, the result follows from a result of Siggers (2010) that shows that a finite structure will have a Taylor polymorphism if and only if it has one of arity 6. Thus, to certify that $\mathbf{B}$ has a Taylor polymorphism, we need only be able to quickly check that a given 6-ary operation on B preserves the relations of $\mathbf{B}$ and is a Taylor operation. ■

Kazda (2014) has shown that the decision problem addressed in Part 1 of Theorem 17 actually lies in the class NP. His algorithm is based on a "local" characterization of congruence modularity and a clever encoding of the problem into an instance of the constraint satisfaction problem over the structure. We note that Siggers' result used in the proof of this theorem has been refined in Kearnes et al. (2014) by reducing the arity of the term from 6 to 4.

5. Learning Lattice Terms

In this section, we prove the hardness of a class of prediction problems that deal with lattices, which will serve as a useful intermediate result on the way to our main hardness result; roughly speaking, the problems studied here involve learning the function induced by an unknown term. When $r \geq 1$ and $\mathcal{L}$ is a finite set of finite lattices, define $\mathcal{C}^r_{\mathrm{TERM}}(\mathcal{L})$ to be the prediction problem containing a pair $(t, (\mathbf{L}, h, c))$ when the following conditions hold: t is a lattice term with $\mathrm{depth}(t) \leq r + r\log(\mathrm{leafsize}(t))$; $\mathbf{L} = (L; \wedge, \vee)$ is a lattice in $\mathcal{L}$; h is an assignment mapping each variable of t to an element of L; c is an element of L; and, $\mathbf{L}, h \models (t \geq c)$, that is, under the assignment h, the term t evaluates to a value greater than or equal to c in $\mathbf{L}$.

Theorem 18 *Suppose that $\mathcal{L}$ is a finite set of finite lattices containing a non-trivial lattice. Then, there exists $r > 1$ such that there exists an oracular pwm-reduction from the prediction problem $\mathcal{C}_{\text{log-MPF}}$ to the prediction problem $\mathcal{C}^r_{\mathrm{TERM}}(\mathcal{L})$.*

It is helpful to first establish this theorem in the case of distributive lattices; the proof uses the fact that each finite distributive lattice can be embedded into a finite power of the two-element lattice.

Lemma 19 *Theorem 18 holds in the case that $\mathcal{L}$ contains only distributive lattices.*

Proof Let $\mathbf{L}_{\{0,1\}}$ denote the two-element lattice with bottom element 0 and top element 1. We first show that there exists a reduction from $\mathcal{C} = \mathcal{C}_{\text{log-MPF}}$ to $\mathcal{C}' = \mathcal{C}^6_{\mathrm{TERM}}(\{\mathbf{L}_{\{0,1\}}\})$. The reduction (f, g, H) is defined as follows. The functions g and f are defined by $g(s, n, \theta) = \theta$ and $f(s, n, h) = (\mathbf{L}_{\{0,1\}}, h, 1)$. The algorithm H, on input $(s, n, (\mathbf{L}_{\{0,1\}}, h, b))$, does the following: if $b = 0$, it outputs 1, and if $b = 1$, it submits h as an oracle query and outputs the result. This reduction is correct, as for each monotone propositional formula θ and assignment h to the variables of θ, it always holds that $(\mathbf{L}_{\{0,1\}}, h, 0) \in \kappa_{\mathcal{C}'}(\theta)$; and, it holds that $(\mathbf{L}_{\{0,1\}}, h, 1) \in \kappa_{\mathcal{C}'}(\theta)$ if and only if h satisfies θ.

Now, suppose that $\mathbf{L}$ is a set of distributive lattices containing a non-trivial lattice. We exhibit a reduction from $\mathcal{C} = \mathcal{C}^6_{\mathrm{TERM}}(\{\mathbf{L}_{\{0,1\}}\})$ to $\mathcal{C}' = \mathcal{C}^6_{\mathrm{TERM}}(\mathcal{L})$, which suffices to give the lemma. It is well-known and straightforward to verify that each finite distributive lattice embeds into a finite power of the lattice $\mathbf{L}_{\{0,1\}}$. Let $D \geq 1$ be a sufficiently large constant so that each $\mathbf{L} \in \mathcal{L}$ has an embedding into $\mathbf{L}_{\{0,1\}}^D$. For each $\mathbf{L} \in \mathcal{L}$, fix $e^{\mathbf{L}}$ to be such an embedding, and let $e_i^{\mathbf{L}}$ be the function that returns the ith coordinate of $e^{\mathbf{L}}$ (for $i = 1, \ldots, D$). Fix $\mathbf{L}^+ \in \mathcal{L}$ to be a non-trivial lattice, and let $\top$ and $\bot$ denote the top and bottom elements of $\mathbf{L}^+$, respectively. Let $e^+ : \{0,1\} \to \{\bot, \top\}$ be the mapping where $e^+(0) = \bot$ and $e^+(1) = \top$. The reduction is (f, g, H), defined as follows. The functions are defined by $g(s, n, t) = t$ and $f(s, n, (\mathbf{L}_{\{0,1\}}, h, c)) = (\mathbf{L}^+, e^+(h), e^+(c))$. The algorithm H, on input $(s, n, (\mathbf{L}, h, c))$, makes D oracle queries: for each $i = 1, \ldots, D$, it submits the query $(\mathbf{L}_{\{0,1\}}, e_i^{\mathbf{L}}(h), e_i^{\mathbf{L}}(c))$ and returns 1 if and only if all oracle calls were answered as 1. The correctness of H follows from the fact that, for any lattice term t, and any triple $(\mathbf{L}, h, c)$, it holds that $\mathbf{L}, h \models t \geq c$ if and only if $\mathbf{L}_{\{0,1\}}^D, e^{\mathbf{L}}(h) \models t \geq e^{\mathbf{L}}(c)$, which in turn is true if and only if for all $i = 1, \ldots, D$, it holds that $\mathbf{L}_{\{0,1\}}, e_i^{\mathbf{L}}(h) \models t \geq e_i^{\mathbf{L}}(c)$. ■

Proof (Theorem 18) By Lemma 19, it suffices to prove the theorem for each such set $\mathbf{L}$ that contains a non-distributive lattice. We prove this by induction on the maximum cardinality of a non-distributive lattice in $\mathbf{L}$. Define $s(x, y, z)$ to be the term $(x \wedge y) \vee (x \wedge z)$, and define $s'(x, y, z)$ to be the term $x \wedge (y \vee z)$. In the scope of this proof, when d and d' are elements of a lattice $\mathbf{L}$ with $d \leq d'$, we use $[d, d']$ to denote the set $\{c \mid d \leq c \leq d'\}$, and we use $\mathbf{L}[d, d']$ to denote the sublattice of $\mathbf{L}$ with universe $[d, d']$. Note that for any elements a, b, c of a lattice $\mathbf{L}$, it always holds that $s(a, b, c) \leq s'(a, b, c)$.

Define $\mathcal{L}^-$ as the set $\{\mathbf{L}[s(a,b,c), s'(a,b,c)] \mid a, b, c \in \mathbf{L} \text{ and } \mathbf{L} \in \mathcal{L}\}$. We will prove that, for any value $r > 1$, it holds that $\mathcal{C}^r_{\text{TERM}}(\mathcal{L}^-)$ has an oracular pwm-reduction to $\mathcal{C}^{r+4}_{\text{TERM}}(\mathcal{L})$. Let us argue that this suffices. Consider a lattice $\mathbf{L} \in \mathcal{L}$. If the lattice $\mathbf{L}$ is distributive, then for any elements $a, b, c \in \mathbf{L}$, it holds that $s(a, b, c) = s'(a, b, c)$ and thus that $\mathbf{L}[s(a, b, c), s'(a, b, c)]$ is a one-element lattice. If the lattice $\mathbf{L}$ is non-distributive, then for any elements $a, b, c \in \mathbf{L}$, if $s'(a, b, c)$ is the top element of $\mathbf{L}$, then a must be equal to the top element of $\mathbf{L}$, which in turn implies that $s(a, b, c) = s'(a, b, c)$. Hence (when $\mathbf{L}$ is non-distributive) each lattice of the form $\mathbf{L}[s(a, b, c), s'(a, b, c)]$ has cardinality strictly smaller than that of $\mathbf{L}$. Now consider two cases. If $\mathcal{L}^-$ contains a non-distributive lattice, then by the argumentation just given and by induction, there exists a value r such that $\mathcal{C}^r_{\text{TERM}}(\mathcal{L}^-)$ admits an oracular pwm-reduction from $\mathcal{C}_{\text{log-MPF}}$, and hence an oracular pwm-reduction from $\mathcal{C}^r_{\text{TERM}}(\mathcal{L}^-)$ to $\mathcal{C}^{r+4}_{\text{TERM}}(\mathcal{L})$ yields the theorem. If $\mathcal{L}^-$ contains only distributive lattices, we claim that $\mathcal{L}^-$ contains a non-trivial lattice, which completes the argument by appeal to Lemma 19. This claim holds because there exists (by assumption) a non-distributive lattice $\mathbf{L} \in \mathcal{L}$; by definition, there exist elements $a, b, c \in \mathbf{L}$ such that $s(a, b, c) \neq s'(a, b, c)$. Hence, the lattice $\mathbf{L}[s(a, b, c), s'(a, b, c)]$ is non-trivial.

It remains to give an oracular pwm-reduction (f, g, H) from $\mathcal{C}^r_{\text{TERM}}(\mathcal{L}^-)$ to $\mathcal{C}^{r+4}_{\text{TERM}}(\mathcal{L})$. First, define $g(r, n, t^-(x_1, \ldots, x_n))$ to be the term $t(z_1, z_2, z_3, x_1, \ldots, x_n)$ defined to be $t^-(x_1^*, \ldots, x_n^*)$, where each x_i^* is defined as the term $(x_i \vee s(z_1, z_2, z_3)) \wedge s'(z_1, z_2, z_3)$. Observe that $\text{depth}(t) \leq \text{depth}(t^-) + 4$. Define $f(r, n, (\mathbf{L}^-, h^-, c^-))$ to be $(\mathbf{L}, h, c^-)$ where $\mathbf{L}$ is a lattice in $\mathbf{L}$ such that there exist $a, b, c \in \mathbf{L}$ with $\mathbf{L}^- = \mathbf{L}[s(a, b, c), s'(a, b, c)]$, and where h is the extension of h^- defined on $\{z_1, z_2, z_3, x_1, \ldots, x_n\}$ where $h(z_1) = a$, $h(z_2) = b$, and $h(z_3) = c$. This f satisfies the needed property, as $\mathbf{L}^-, h^- \models t^- \geq c^-$ holds if and only if $\mathbf{L}, h^- \models t^- \geq c^-$ holds; this latter condition is equivalent to $\mathbf{L}, h \models t \geq c^-$, as $h(x_i^*)$ is equal to $h^-(x_i)$ for each i. Define the algorithm H on $(r, n, (\mathbf{L}, h, d))$ to perform the following. Let $\mathbf{L}^-$ be the lattice $\mathbf{L}[s(h(z_1), h(z_2), h(z_3)), s'(h(z_1), h(z_2), h(z_3))]$. Define h^- on $\{x_1, \ldots, x_n\}$ by $h^-(x_i) = (h(x_i) \vee s(h(z_1), h(z_2), h(z_3))) \wedge s'(h(z_1), h(z_2), h(z_3))$. Set D^- to be the set $\{d^- \in \mathbf{L}^- \mid d^- \geq d\}$. The algorithm H makes, for each $d^- \in D^-$, the oracle query $(\mathbf{L}^-, h^-, d^-)$, and returns 1 if and only if at least one of the oracle responses was 1. Let us discuss why this algorithm satisfies the desired property. It is readily verified that, when t and t^- are terms with $g(s, n, t^-(x_1, \ldots, x_n)) = t(z_1, z_2, z_3, x_1, \ldots, x_n)$ and $(\mathbf{L}, h, d)$ is a triple, that $\mathbf{L}, h \models t \geq d$ if and only if $\mathbf{L}, h \models t^-(x_1^*, \ldots, x_n^*) \geq d$ if and only if $\mathbf{L}, h^- \models t^-(x_1, \ldots, x_n) \geq d$. Since all values in the image of h^- are in $\mathbf{L}^-$, the last condition $\mathbf{L}, h^- \models t^-(x_1, \ldots, x_n) \geq d$ holds if and only if there exists $d^- \in D^-$ such that $\mathbf{L}^-, h^- \models t^-(x_1, \ldots, x_n) \geq d^-$. ■

6. Hardness From Non-Congruence Modularity

In this section, we make use of 2-sorted relational structures. We thus first give an introduction to these. A 2-*sorted signature* is a finite set of relation symbols, where each symbol has an associated arity which is a string over $\{1, 2\}$. Below, the only 2-sorted signature to be used is $\{R\}$, where the symbol R has arity 122. A 2-*sorted relational structure* over a 2-sorted signature σ consists of two finite sets B_1, B_2 called the *first* and *second* universe, respectively, and, for each relation symbol $S \in \sigma$, a relation $S^{\mathbf{B}}$ that is a subset of $B_{w_1} \times \cdots \times B_{w_k}$, where $w_1 \ldots w_k$ is the arity of S. A conjunctive query on a 2-sorted signature is defined similarly to a conjunctive query on a usual relational signature, but with the differences that each variable has an associated sort (1 or 2), equality of variables may only be written between variables of the same sort, and when $S(v_1, \ldots, v_k)$ is a predicate application, letting $s(v_i)$ denote the sort of v_i, it must be that $s(v_1) \ldots s(v_k)$ is equal to the arity of S.

We now turn to the proper content of this section. Let A be a set. When θ and θ' are binary relations on A, we use $\theta \circ \theta'$ to denote their relational product. We use $\mathrm{Eq}(A)$ to denote the lattice of equivalence relations on A, and we use $0_A = \{(a, a) \mid a \in A\}$ and $1_A = A^2$ to denote the bottom and top elements of $\mathrm{Eq}(A)$, respectively. We define a *pentagon* to be a finite relational structure $\mathbf{P}$ over the signature $\{\alpha, \beta, \gamma\}$ containing three binary relation symbols such that $\alpha^{\mathbf{P}}$, $\beta^{\mathbf{P}}$, and $\gamma^{\mathbf{P}}$ are equivalence relations on P, and the following conditions hold in $\mathrm{Eq}(P)$: $\alpha^{\mathbf{P}} \leq \beta^{\mathbf{P}}$, $\beta^{\mathbf{P}} \wedge \gamma^{\mathbf{P}} = 0_P$, $\beta^{\mathbf{P}} \circ \gamma^{\mathbf{P}} = 1_P$, and $\alpha^{\mathbf{P}} \vee \gamma^{\mathbf{P}} = 1_P$. The universe P of a pentagon $\mathbf{P}$ can be naturally decomposed as a direct product $P = B \times C$ in such a way that $\beta^{\mathbf{P}}$ and $\gamma^{\mathbf{P}}$ are the kernels of the projections of P onto B and C, respectively. Then, via the equivalence relation $\alpha^{\mathbf{P}}$, each element $b \in B$ induces an equivalence relation $\alpha_b^{\mathbf{P}} = \{(c, c') \in C \times C \mid ((b, c), (b, c')) \in \alpha^{\mathbf{P}}\}$ on C. For each pentagon $\mathbf{P}$, we define $\mathbf{L}(\mathbf{P})$ to be the lattice which is the sublattice of $\mathrm{Eq}(C)$ generated by the equivalence relations $\alpha_b^{\mathbf{P}}$ (over $b \in B$); we extend this operator $\mathbf{L}(\cdot)$ to sets of pentagons in the natural fashion.

To each pentagon $\mathbf{P}$, we associate a 2-sorted relational structure, denoted by $\mathbf{P}_2$, which has $B_{\mathbf{P}}$ and $C_{\mathbf{P}}$ as first and second universe, respectively; here, $B_{\mathbf{P}}$ and $C_{\mathbf{P}}$ denote the sets in the decomposition of the universe P as described above. The structure $\mathbf{P}_2$ is defined on signature $\{R\}$ and has $R^{\mathbf{P}_2} = \{(b, c, c') \in B_{\mathbf{P}} \times C_{\mathbf{P}} \times C_{\mathbf{P}} \mid (c, c') \in \alpha_b^{\mathbf{P}}\}$. The definition of $\mathbf{P}_2$ comes from (Bova et al., 2013). In forming conjunctive queries over this signature $\{R\}$

each variable has a sort (first or second) associated with each variable; an atom $R(x, y, y')$ may be formed if x is of the first sort and y and y' are of the second sort. When $\mathcal{P}$ is a set of pentagons, we define the prediction problem $\mathcal{C}_{\text{CQ-2-PENT}}(\mathcal{P})$ to be the set

$$\{(\phi(V_1, V_2), (\mathbf{P}, (h_1, h_2))) \mid \mathbf{P} \in \mathcal{P} \text{ and } h_1 : V_1 \to B_{\mathbf{P}},\ h_2 : V_2 \to C_{\mathbf{P}} \text{ such that } \mathbf{P}_2, h_1, h_2 \models \phi\ \}.$$

Here, $\phi(V_1, V_2)$ denotes a conjunctive query over the signature $\{R\}$ with V_1 a set of variables of the first sort and V_2 a set of the second sort.

Theorem 20 *For $\mathcal{P}$ a finite set of pentagons, there exists an oracular pwm-reduction from the prediction problem $\mathcal{C}^r_{\text{TERM}}(\mathbf{L}(\mathcal{P}))$ for any $r > 1$ to the prediction problem $\mathcal{C}_{\text{CQ-2-PENT}}(\mathcal{P})$.*

Proof We make use of a version of a construction presented in the proof of Theorem 10 from Bova et al. (2013), which produces a 2-sorted conjunctive query $\phi_t(x_1, \ldots, x_m, y, y')$ over the signature $\{R\}$ from a lattice term $t(x_1, \ldots, x_m)$, where in ϕ_t the variables x_i are of sort 1 and the variables y and y' are of sort 2. The construction has the property that if $\mathbf{P} \in \mathcal{P}$, then for all $b_1, \ldots, b_m \in B_{\mathbf{P}}$ and for all c, $c' \in C_{\mathbf{P}}$, $\phi_t(b_1, \ldots, b_m, c, c')$ holds in $\mathbf{P}_2$ if and only if the pair (c, c') is in the equivalence relation given by $t^{\mathbf{L}(\mathbf{P})}(\alpha^{\mathbf{P}}_{b_1}, \ldots, \alpha^{\mathbf{P}}_{b_m})$. Let us specify the version of the construction used here.

- If $t = x_i$, then $\phi_t(x_1, \ldots, x_m, y, y') = R(x_i, y, y')$.
- If $t = t_1 \wedge t_2$, then $\phi_t(x_1, \ldots, x_m, y, y') = \phi_{t_1}(x_1, \ldots, x_m, y, y') \wedge \phi_{t_2}t(x_1, \ldots, x_m, y, y')$.
- If $t = t_1 \vee t_2$, then set M to be the maximum size of a second universe $C_{\mathbf{P}}$ over all pentagons $\mathbf{P} \in \mathcal{P}$. Let $z_{0,2}$ and $z_{i,j}$, where $i = 1, \ldots, M$ and $j = 1, 2$, be variables of the second sort, and identify $y = z_{0,2}$ and $y' = z_{M,2}$. Then $\phi_t(x_1, \ldots, x_m, y, y')$ is defined as the formula obtained by existentially quantifying the variables $z_{i,j}$ (other than y and y') before the conjunction
$$\bigwedge_{i=1}^{M} (\phi_{t_1}(x_1, \ldots, x_m, z_{i-1,2}, z_{i,1}) \wedge \phi_{t_2}(x_1, \ldots, x_m, z_{i,1}, z_{i,2})).$$

Note that in each of the latter two cases, the size $|\phi_t|$ of the created formula ϕ_t has size bounded above by a constant times $|\phi_{t_1}| + |\phi_{t_2}|$. Hence, the size of ϕ_t will be polynomial in that of t—when t has logarithmic depth, which will be the case in our application here.

To define the g component of the reduction, the construction just given is not applied directly to a lattice term t. Instead, we first start with a fixed lattice term $s(x_1, \ldots, x_q)$ with the property that for all $\mathbf{P} \in \mathcal{P}$ and all $\delta \in \mathbf{L}(\mathbf{P})$ there are $b_i \in B_{\mathbf{P}}$, for $1 \leq i \leq q$, so that $\delta = s^{\mathbf{L}(\mathbf{P})}(\alpha^{\mathbf{P}}_{b_1}, \ldots, \alpha^{\mathbf{P}}_{b_q})$. We let $\omega(\delta)$ be some sequence $(b_1, \ldots, b_q)$ for which this equality holds. The existence of such a term follows from the fact that $\mathcal{P}$ is a finite set of finite pentagons and that in each pentagon $\mathbf{P}$, the equivalence relations $\alpha^{\mathbf{P}}_b$ generate the lattice $\mathbf{L}(\mathbf{P})$. For future reference, let u be an integer such that $|\mathbf{P}| \leq u$ for all $\mathbf{P} \in \mathcal{P}$. For any lattice term $t(x_1, \ldots, x_m)$, we define $t \star s$ to be the mq-ary lattice term

$$t(s(x_{11}, \ldots, x_{1q}), \ldots, s(x_{m1}, \ldots, x_{mq})).$$

Let us now begin to describe the reduction; the parameters s and n (as described in the definition of oracular pwm-reduction) do not play a role, and we omit their mention. For each lattice term $t(x_1, \ldots, x_m)$, define $g(t)$ to be the following conjunctive query over the signature $\{R\}$:

$$\bigwedge_{1 \leq i \leq u^2} \phi_{t \star s}(x_{11}, x_{12}, \ldots, x_{mq}, y_i, y_i').$$

For $\mathbf{P} \in \mathcal{P}$, an m-tuple $h = (\delta_1, \ldots, \delta_m)$ over $\mathbf{L}(\mathbf{P})$, and $\theta \in \mathbf{L}(\mathbf{P})$, define the function $f((\mathbf{L}(\mathbf{P}), h, \theta))$ to be $\mathbf{P}$ along with the assignment of the variables $(x_{11}, \ldots, x_{mq})$ to the concatenation of the q-tuples $\omega(\delta_i)$, for $1 \leq i \leq m$. The pairs of variables (y_i, y_i'), for $1 \leq i \leq u^2$, are assigned to pairs $(c, c') \in \theta$ so that each pair in θ is in the range of this assignment. By the choice of u, this is always possible to arrange. Note that $f((\mathbf{L}(\mathbf{P}), h, \theta))$ can be computed in time bounded by some polynomial in m.

It follows from the claim made about the construction ϕ_t in the first paragraph of this proof, and from the properties of the lattice term s that for all lattice terms $t(x_1, \ldots, x_m)$, if $\mathbf{P} \in \mathcal{P}$, $h = (\delta_1, \ldots, \delta_m) \in \mathbf{L}(\mathbf{P})^m$, and $\theta \in \mathbf{L}(\mathbf{P})$ then $t^{\mathbf{L}(\mathbf{P})}(\delta_1, \ldots, \delta_m) \geq \theta$ if and only if $\mathbf{P}_2$ satisfies the conjunctive query $g(t)$ under the assignments given by $f((\mathbf{L}(\mathbf{P}), h, \theta))$.

To complete the definition of our reduction from $\mathcal{C}^r_{\text{TERM}}(\mathbf{L}(\mathcal{P}))$ to $\mathcal{C}_{\text{CQ-2-PENT}}(\mathcal{P})$, define H to be the algorithm that when given $\mathbf{P} \in \mathcal{P}$ and h_1 and h_2 tuples over $B_{\mathbf{P}}$ and $C_{\mathbf{P}}$ respectively, will reject the input if h_1 is not an mq-tuple for some integer m or if h_2 is a tuple which is not of length $2u^2$. Otherwise, with $h_1 = (b_{11}, \ldots, b_{mq})$ and $h_2 = (c_1, c_1', \ldots, c_{u^2}, c_{u^2}')$, construct in $\mathbf{L}(\mathbf{P})$ the elements $\delta_i = s^{\mathbf{L}(\mathbf{P})}(\alpha^{\mathbf{P}}_{b_{i1}}, \ldots, \alpha^{\mathbf{P}}_{b_{iq}})$ and find the smallest element $\theta \in \mathbf{L}(\mathbf{P})$ such that $(c_i, c_i') \in \theta$ for all $1 \leq i \leq u^2$. For any lattice term t, the algorithm will return the result of testing if in $\mathbf{P}$, $t \geq \theta$ under the assignment $(\delta_1, \ldots, \delta_m)$. From the properties of g noted earlier, it follows that in $\mathbf{P}$, $t \geq \theta$ under this assignment if and only if $g(t)$ is true in $\mathbf{P}_2$ under the assignment (h_1, h_2). The run time of H can be bounded by a polynomial in the lengths of h_1 and h_2. ■

Lemma 21 *(Bova et al., 2013) Let* $\mathbf{B}$ *be a finite relational structure such that* $\mathcal{V}(\mathbb{A}(\mathbf{B}))$ *is not congruence modular. There exists a finite relational structure* $\mathbf{A}$ *defined on a signature including three binary relation symbols* α*,* β*, and* γ *which is compatible with an algebra in* $\mathcal{V}(\mathbb{A}(\mathbf{B}))$*, such that the following hold:*

- *There exists a finite set* $\mathcal{P}$ *of pentagons where for each* $\mathbf{P} \in \mathcal{P}$*, the universe* P *of* $\mathbf{P}$ *is a subset of* A*, and it holds that* $\alpha^{\mathbf{P}} = \alpha^{\mathbf{A}} \cap P^2$*,* $\beta^{\mathbf{P}} = \beta^{\mathbf{A}} \cap P^2$*, and* $\gamma^{\mathbf{P}} = \gamma^{\mathbf{A}} \cap P^2$*. Moreover, the set* $\mathbf{L}(\mathcal{P})$ *contains a non-trivial lattice.*
- *For each* $k \geq 1$*, there exists a relation* $D_k \subseteq A^k$ *which is cq-definable over* $\mathbf{A}$ *such that for any elements* $a_1, \ldots, a_k \in A$*, the tuple* $(a_1, \ldots, a_k)$ *is in* D_k *if and only if there exists a* $\mathbf{P} \in \mathcal{P}$ *such that all of the elements* $a_1, \ldots, a_k$ *are contained in the universe* P *of* $\mathbf{P}$*. In addition, there exists an algorithm that computes a cq-definition of* D_k *(over* $\mathbf{A}$*) in polynomial time, when given* k *as input.*

In the definition of the set $\mathcal{P}$ we may assume that if $\mathbf{P}$, $\mathbf{P}'$ are members, then $P \nsubseteq P'$. This additional property can be arranged by only including in $\mathcal{P}$ those pentagons whose universes are maximal with respect to inclusion. Doing so will not change the other properties listed in the previous lemma.

Theorem 22 *Let* $\mathbf{A}$ *be a finite relational structure satisfying the conditions described in Lemma 21, and let* $\mathcal{P}$ *be the set of pentagons described there. There exists an oracular pwm-algorithm from* $\mathcal{C}_{\text{CQ-2-PENT}}(\mathcal{P})$ *to* $\mathcal{C}_{\text{CQ}}(\mathbf{A})$*.*

Essentially, Theorem 22 is proved in the following way. In order to translate a 2-sorted conjunctive query ϕ over pentagons to a conjunctive query ϕ' over $\mathbf{A}$, the relations β and γ are used to simulate the two sorts, and the relation α is used to simulate the behavior of the relation R. Also, in the resulting conjunctive query ϕ', all of the variables are related by the relation D_U (where U is the total number of variables), effectively localizing ϕ' to the pentagons found in the set $\mathcal{P}$.

Proof Let M be a natural number such that $|P| \leq M$ for all $\mathbf{P} \in \mathcal{P}$. For ϕ a 2-sorted conjunctive query over the signature $\{R\}$, let $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$ be the variables of the first and second sort, respectively, that appear in ϕ. Assume that the free variables of ϕ are $\{x_1, \ldots, x_{n'}\}$ and $\{y_1, \ldots, y_{m'}\}$ where $n' \leq n$ and $m' \leq m$. We construct a conjunctive query ϕ' from ϕ over the signature $\{\alpha, \beta, \gamma\}$ along the lines of the construction found in the proof of Theorem 7 in Bova et al. (2013); but, instead of adding the conjunct

$$\Delta_{n+m+k}(x'_1, \ldots, x'_n, y'_1, \ldots, y'_m, w'_1, \ldots, w'_k),$$

in the construction of ϕ' we add the conjunct

$$D_{n+m+k+M}(x'_1, \ldots, x'_n, y'_1, \ldots, y'_m, w'_1, \ldots, w'_k, v'_1, \ldots, v'_M),$$

where the v'_i's are fresh variables and the relation D is provided by the previous lemma. The resulting formula ϕ' will have as free variables

$$\{x'_1, \ldots, x'_{n'}\} \cup \{y'_1, \ldots, y'_{m'}\} \cup \{v'_1, \ldots, v'_M\}.$$

In giving the oracular pwm-reduction, the parameters s and n (as described in the definition of oracular pwm-reduction) do not play a role, and we omit their mention. To construct an oracular pwm-reduction (f, g, H) from $\mathcal{C}_{\text{CQ-2-PENT}}(\mathcal{P})$ to $\mathcal{C}_{\text{CQ}}(\mathbf{A})$ we define g to be the function that maps the 2-sorted conjunctive query ϕ to the conjunctive query ϕ'. As in the original construction from Bova et al. (2013) it follows that the size of $g(\phi)$ can be bounded by a polynomial in the size of ϕ. A key point, as appears in the statement of Lemma 21, is that a cq-definition of the relation D_k can be constructed in time bounded by some polynomial in k.

Given $\mathbf{P} \in \mathcal{P}$, let b^* be some fixed member of $B_\mathbf{P}$ and c^* some fixed member of $C_\mathbf{P}$ and let $(p_1, \ldots, p_M)$ be some listing of the elements of P. For the purposes of this discussion we regard P as being equal to the set $B_\mathbf{P} \times C_\mathbf{P}$, though strictly speaking, P is a subset of A. For assignments $h_1 = (b_1, \ldots, b_n) \in B^n_\mathbf{P}$ and $h_2 = (c_1, \ldots, c_m) \in C^m_\mathbf{P}$, define $f((\mathbf{P}, (h_1, h_2)))$ to be the tuple

$$((b_1, c^*), \ldots, (b_n, c^*), (b^*, c_1), \ldots, (b^*, c_m), p_1, \ldots, p_M).$$

Clearly $f((\mathbf{P}, (h_1, h_2)))$ can be computed in time bounded by a polynomial in the sizes of h_1 and h_2.

For ϕ a 2-sorted conjunctive query over the signature $\{R\}$, $\mathbf{P} \in \mathcal{P}$, and (h_1, h_2) a sorted assignment of the free variables of ϕ to $\mathbf{P}_2$, it follows from Claim 4 in the proof of Theorem 7 from Bova et al. (2013) (and referred to as just Claim 4 in the remainder of this proof) that if ϕ is true in $\mathbf{P}_2$ under the sorted assignment (h_1, h_2) then in $\mathbf{A}$, the formula $g(\phi) = \phi'$ is true under the assignment $f((\mathbf{P}, (h_1, h_2)))$. Claim 4 applies in this situation, since, using terminology from it, the assignments (h_1, h_2) and $f((\mathbf{P}, (h_1, h_2)))$, when restricted to its first $n + m$ components, *match*. The addition of the last M components to $f((\mathbf{P}, (h_1, h_2)))$ do not affect the satisfaction of ϕ', since they only appear in the D-relation conjunct of ϕ' and all of the elements lie in the set P and so, together will satisfy the D-relation that appears in ϕ'.

Conversely, if ϕ' is satisfied in $\mathbf{A}$ under the assignment $f((\mathbf{P}, (h_1, h_2)))$ then by Claim 4, there is some $\mathbf{P}' \in \mathcal{P}$ such that in $\mathbf{P}'_2$, the formula ϕ is satisfied under the assignment (h_1, h_2). By Lemma 21, all of the elements of A that appear in $f((\mathbf{P}, (h_1, h_2)))$ must all lie in $\mathbf{P}'$ since they jointly satisfy a D-relation. But by our modification of the set $\mathcal{P}$ and the fact that $(p_1, \ldots, p_M)$ is a listing of the elements of P, it follows that $\mathbf{P} = \mathbf{P}'$. Thus we have established that $\mathbf{P}$ satisfies ϕ under the assignment (h_1, h_2) if and only if $\mathbf{A}$ satisfies $g(\phi)$ under the assignment $f((\mathbf{P}, (h_1, h_2)))$.

To complete the construction of the oracular pwm-reduction, if we are given an assignment

$$h = (b_1, \ldots, b_n, c_1, \ldots, c_m, p_1, \ldots, p_M)$$

of elements from A, the algorithm H will reject h if there is no $\mathbf{P} \in \mathcal{P}$ that contains all of the elements that appear in h. Otherwise, let $\mathcal{P}_h$ be the set of pentagons in $\mathcal{P}$ that contains this set of elements. If $\phi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ is a 2-sorted conjunctive query over the signature $\{R\}$, then by Claim 4, $g(\phi) = \phi'$ will be satisfied in $\mathbf{A}$ under the assignment h if and only if there is at least one $\mathbf{P} \in \mathcal{P}_h$ such that ϕ will be satisfied in $\mathbf{P}$ under the sorted assignment (h_1, h_2) that matches the assignment $(b_1, \ldots, b_n, c_1, \ldots, c_m)$ in P. By this we mean that $h_1 = (b'_1, \ldots, b'_n)$ and $h_2 = (c'_1, \ldots, c'_m)$, where, regarding the b_i and c_j as elements of $P = B_{\mathbf{P}} \times C_{\mathbf{P}}$, we have $b_i = (b'_i, v_i)$ and $c_j = (u_j, c'_j)$ for some elements v_i and u_j.

So, given the assignment h, and a conjunctive query ϕ, the algorithm H will, for each $\mathbf{P} \in \mathcal{P}_h$ compute the matching assignment (h_1, h_2) and query whether ϕ is satisfied in $\mathbf{P}$ under this assignment. If any of the queries are true, then H will return true to the query testing for the satisfaction of ϕ' in $\mathbf{A}$ under the assignment h. Otherwise, H will return false. Since the number of oracle calls that H will make is bounded by the fixed size of $\mathcal{P}$ and since each matching assignment can be quickly computed, the run time of H can be bounded by a polynomial in the size of h. ■

7. Hardness From the Absence of Taylor Polymorphisms

In this section we prove the following theorem.

Theorem 23 *If* $\mathbf{B}$ *is a finite relational structure that does not have a Taylor polymorphism then there is an oracular pwm-reduction from* $\mathcal{C}_{\exists\mathrm{CIRC}}$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$*.*

Our proof is based on the proof of Theorem 4 from Bova et al. (2013) and makes use of the fact that a finite relational structure $\mathbf{B}$ fails to have a Taylor polymorphism if and only if the variety generated by $\mathbb{A}(\mathbf{B})$ admits a certain kind of "bad" local behaviour. The theory developed by Hobby and McKenzie (Hobby and McKenzie, 1988), known as tame congruence theory, provides a way to analyze the local structure of finite algebras and identifies fives different types of local behaviour that a finite algebra can exhibit. These five types are: (1) the unary type, (2) the affine type, (3) the boolean type, (4) the lattice type, and (5) the semilattice type. A surprisingly useful invariant of a finite algebra $\mathbb{A}$, called the typeset of $\mathbb{A}$, is the set of types of local behaviour that $\mathbb{A}$ exhibits. The notion of a typeset can be extended to a variety $\mathcal{V}$ (or to any collection of algebras) by taking it to be the union of the typesets of its finite members.

Tame congruence theory establishes, for varieties $\mathcal{V}$ generated by a finite algebra, strong connections between the local behaviour of the finite members of $\mathcal{V}$ and global properties of $\mathcal{V}$. Chapter 9 of Hobby and McKenzie (1988) presents several theorems of this sort, including the following:

Theorem 24 *(Hobby and McKenzie, 1988, see Theorem 9.6) Let* $\mathbb{A}$ *be a finite algebra. The following are equivalent:*

- $\mathbb{A}$ *has a Taylor term,*
- *The typeset of* $\mathcal{V}(\mathbb{A})$ *omits the unary type.*

In the proof of Theorem 23 we will use this result to conclude that in the absence of a Taylor polymorphism, some finite member of the variety generated by $\mathbb{A}(\mathbf{B})$ will locally behave as a unary algebra, i.e., an algebra whose basic operations depend on at most one variable. As detailed in the proof, this will provide a way to produce an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{\mathrm{3SAT}})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B})$.

Recall the structure $\mathbf{B}_{\mathrm{3SAT}}$ from Example 1. We will make use of the following fact, which follows immediately from the standard assignment-preserving conversion from a Boolean circuit to a 3-SAT formula where existential quantification may be used.

Proposition 25 *There exists an oracular pwm-reduction from* $\mathcal{C}_{\exists\mathrm{CIRC}}$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{\mathrm{3SAT}})$*.*

We also need the following fact.

Proposition 26 *There is a finite relational structure* $\mathbf{C}$ *with domain* $\{0,1\}$ *such that each relation* $R \subseteq \{0,1\}^n$ *of* $\mathbf{C}$ *contains the constant tuples* $(0,0,\dots,0)$ *and* $(1,1,\dots,1)$ *and such that there is an oracular pwm-reduction from* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{\mathrm{3SAT}})$ *to* $\mathcal{C}_{\mathrm{CQ}}(\mathbf{C})$*.*

Proof We apply the construction from Lemma 4 from Bova et al. (2012) to the structure $\mathbf{B}_{\mathrm{3SAT}}$ to obtain the desired structure $\mathbf{C}$. For each ternary relation R of $\mathbf{B}_{\mathrm{3SAT}}$, $\mathbf{C}$ will have a 5-ary relation R' such that

$$R' = \{(0,1,a,b,c) \mid (a,b,c) \in R\} \cup \{(0,0,0,0,0),(1,1,1,1,1)\}.$$

For any conjunctive query $\phi(x_1, \ldots, x_n)$ of $\mathbf{B}_{3\mathrm{SAT}}$ we construct a corresponding conjunctive query $\phi'(w_0, w_1, x_1, \ldots, x_n)$ of $\mathbf{C}$ by replacing each occurrence of a relation $R(x, y, z)$ in ϕ with $R'(w_0, w_1, x, y, z)$. It is elementary to show that for any n-tuple $\bar{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$, $\phi(\bar{b})$ holds in $\mathbf{B}_{3\mathrm{SAT}}$ if and only if $\phi'(0, 1, \bar{b})$ holds in $\mathbf{C}$. Furthermore if each variable x_i appears in at least one conjunct of ϕ, then the only other $(n+2)$-tuples that satisfy ϕ' are the constant tuples $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$. From this it follows that there is an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{B}_{3\mathrm{SAT}})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{C})$. ■

Proof [Proof of Theorem 23] Suppose that $\mathbf{B}$ is a finite relational structure that does not have a Taylor polymorphism. Then by Lemma 9.4 and Theorem 9.6 from Hobby and McKenzie (1988), the variety $\mathcal{V}$ generated by $\mathbb{A}(\mathbf{B})$ admits the unary type. Let $\mathbf{C}$ be a structure as described in Lemma 26. We argue that there is an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{C})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$, where $\mathbf{A}$ is the finite structure from Lemma 1 from Bova et al. (2013). Since $\mathbf{A}$ is compatible with a finite algebra $\mathbb{A} \in \mathcal{V}$ then the Theorem follows from Propositions 3 and 5.

In addition to providing the structure $\mathbf{A}$ and the algebra $\mathbb{A}$, Lemma 1 also gives a polynomial-time construction that on input a conjunctive query ϕ over $\mathbf{C}$, produces a conjunctive query ϕ' over $\mathbf{A}$ that has the same sets of free and bound variables that ϕ has. If $X = \{x_1, \ldots, x_m\}$ are the free variables of ϕ (and of ϕ') then Proposition 2 from Bova et al. (2013) proves that for any assignment $g : X \to \{0,1\}$, $\mathbf{C}, g \models \phi$ if and only if $\mathbf{A}, g \models \phi'$. Conversely, for any $g : X \to A$, the equivalence of the following two statements is established in the proof of Proposition 2:

- $\mathbf{A}, g \models \phi'$ and
- $\mathbf{A}, g \models E_m$ and for every unary polynomial $p(x)$ of $\mathbb{A}$, if $p(g(x)) \in \{0,1\}$ for all $x \in X$, then $\mathbf{C}, p \circ g \models \phi$.

We use the above to produce an oracular pwm-reduction from $\mathcal{C}_{\mathrm{CQ}}(\mathbf{C})$ to $\mathcal{C}_{\mathrm{CQ}}(\mathbf{A})$ as follows. We need to design an algorithm H such that on input a conjunctive query $\phi(x_1, \ldots, x_m)$ over $\mathbf{C}$ and an assignment $g : \{x_1 \ldots, x_m\} \to A$, decides if $\mathbb{A}, g \models \phi'$. Using the equivalence, it suffices to have H decide if $\mathbf{A}, g \models E_m$ and, for each unary polynomial $p(x)$ of $\mathbb{A}$ with $p(g(x_i)) \in \{0,1\}$ for all $1 \leq i \leq m$, if $\mathbf{C}, p \circ g \models \phi$. From Lemma 1 from Bova et al. (2013) it follows that determining if $\mathbf{A}, g \models E_m$ can be checked in time bounded by a polynomial in m, since for $m > k = |A|$, E_m is equal to the conjunction of $\binom{m}{k}$ instances of E_k, one for each k-element subset of variables from $\{x_1, \ldots, x_m\}$.

Since $\mathbb{A}$ is a finite algebra, then the set of unary polynomials $p(x)$ of $\mathbb{A}$ is also finite (of size bounded by k^k). For each such polynomial $p(x)$, H checks to see if $p(g(x_i)) \in \{0,1\}$ for all $1 \leq i \leq m$. For such a polynomial, H queries whether $\mathbf{C}, p \circ g \models \phi$. If any of these queries are false, then H returns false, otherwise, H returns true. Then $\mathbf{A}, g \models \phi'$ if and only if H returns true on input ϕ and g. The number of queries that H makes is bounded by the constant k^k and the run-time of H is bounded by m and the size of ϕ. ■

8. Open Issues

Let us conclude by highlighting two open issues. First, we ask if one can provide evidence, even of the conditional sort, that the dichotomy of Theorem 15 is a true dichotomy. Second, it would be desirable to pin down the exact complexity of the meta-problems of deciding whether or not an input relational structure satisfies each of the dichotomy conditions; we exhibited upper bounds on these meta-problems in Theorem 17 and the surrounding discussion.

Acknowledgments

An extended abstract of this work appeared in the Proceedings of the 28th Conference on Learning Theory (Chen and Valeriote, 2015). The second author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada.

References

Eric Allender, Michael Bauland, Neil Immerman, Henning Schnoor, and Heribert Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. *Journal of Computer and System Sciences*, 75(4):245–254, 2009.

Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, April 1988. ISSN 0885-6125. doi: 10.1023/A:1022821128753.

Dana Angluin and Michael Kharitonov. When won't membership queries help? *Journal of Computer and System Sciences*, 50(2):336 – 355, 1995. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1995.1026.

Dana Angluin, Michael Frazier, and Leonard Pitt. Learning conjunctions of Horn clauses. *Mach. Learn.*, 9(2-3):147–164, July 1992. ISSN 0885-6125. doi: 10.1007/BF00992675.

Marta Arias and Roni Khardon. Learning closed Horn expressions. *Information and Computation*, 178(1):214 – 240, 2002. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.2002.3162.

Libor Barto. Finitely related algebras in congruence modular varieties have few subpowers. *Journal of the European Mathematical Society*, 20, 04 2018. doi: 10.4171/JEMS/790.

Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, January 2014. ISSN 0004-5411. doi: 10.1145/2556646.

Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. ISBN 978-3-95977-003-3.

Joel Berman, Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Varieties with few subalgebras of powers. *Transactions of the American Mathematical Society*, 362(3):1445–1473, 2010.

Arnab Bhattacharyya and Yuichi Yoshida. An algebraic characterization of testable Boolean CSPs. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 123–134, 2013.

Maria Luisa Bonet and Samuel Buss. Size-depth tradeoffs for Boolean formulae. *Information Processing Letters*, 49(3):151 – 155, 1994. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(94)90093-0.

Simone Bova, Hubie Chen, and Matthew Valeriote. On the expression complexity of equivalence and isomorphism of primitive positive formulas. *Theory Comput. Syst.*, 50(2):329–353, 2012.

Simone Bova, Hubie Chen, and Matthew Valeriote. Generic expression hardness results for primitive positive formula comparison. *Inf. Comput.*, 222:108–120, 2013.

Nader Bshouty. Exact learning from membership queries: Some techniques, results and new directions. In *Algorithmic Learning Theory - 24th International Conference, ALT 2013, Singapore, October 6-9, 2013. Proceedings*, pages 33–52, 2013.

Nader Bshouty, Jeffrey Jackson, and Christino Tamon. Exploring learnability between exact and PAC. *J. Comput. Syst. Sci.*, 70(4):471–484, 2005.

Andrei Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34:1–34:41, October 2013. ISSN 0004-5411. doi: 10.1145/2528400.

Andrei Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017. ISBN 978-1-5386-3464-6. doi: 10.1109/FOCS.2017.37.

Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

Andrei Bulatov, Hubie Chen, and Victor Dalmau. Learning intersection-closed classes with signatures. *Theoretical Computer Science*, 382(3):209–220, 2007.

Stanley Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Springer, 1981.

Hubie Chen. Meditations on quantified constraint satisfaction. In Robert Constable and Alexandra Silva, editors, *Logic and Program Semantics*, volume 7230 of *Lecture Notes in Computer Science*, pages 35–49. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-29484-6.

Hubie Chen and Benoit Larose. Asking the metaquestions in constraint tractability. *ACM Trans. Comput. Theory*, 9(3):11:1–11:27, October 2017. ISSN 1942-3454. doi: 10.1145/3134757.

Hubie Chen and Matthew Valeriote. Learnability of solutions to conjunctive queries: The full dichotomy. In Peter Grnwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 326–337, Paris, France, 03–06 Jul 2015. PMLR.

Hubie Chen, Matthew Valeriote, and Yuichi Yoshida. Testing assignments to constraint satisfaction problems. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 525–534, Oct 2016. doi: 10.1109/FOCS.2016.63.

Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2001.

Victor Dalmau. A dichotomy theorem for learning quantified Boolean formulas. *Machine Learning*, 35(3):207–224, 1999.

Victor Dalmau and Peter Jeavons. Learnability of quantified formulas. *Theor. Comput. Sci.*, 306(1-3):485–511, 2003.

Tomás Feder and Moshe Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.

Ralph Freese and Ralph McKenzie. *Commutator Theory for Congruence Modular Varieties*, volume 125 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1987.

Ralph Freese and Matthew Valeriote. On the complexity of some Maltsev conditions. *Internat. J. Algebra Comput.*, 19(1):41–77, 2009. ISSN 0218-1967.

David Hobby and Ralph McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics.* American Mathematical Society, 1988.

Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

Jeffrey Jackson and Rocco Servedio. On learning random DNF formulas under the uniform distribution. *Theory of Computing*, 2(1):147–172, 2006.

Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

Alexandr Kazda. Personal communication, 2014.

Keith Kearnes, Petar Marković, and Ralph McKenzie. Optimal strong Mal'cev conditions for omitting type 1 in locally finite varieties. *Algebra universalis*, 72(1):91–100, aug 2014. ISSN 0002-5240, 1420-8911. doi: 10.1007/s00012-014-0289-9.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, Apr 1988.

Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3-4):463–489, 2008. ISSN 0002-5240. doi: 10.1007/s00012-008-2122-9.

Ralph McKenzie, George McNulty, and Walter Taylor. *Algebras, Lattices, Varieties, vol. 1.* Wadsworth & Brooks/Cole, 1987.

Leonard Pitt and Manfred Warmuth. Prediction-preserving reducibility. *J. Comput. Syst. Sci.*, 41(3):430–467, 1990.

Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 245–254, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-047-0. doi: 10.1145/1374376.1374414.

Thomas Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM. doi: 10.1145/800133.804350.

Mark Siggers. A strong Mal'cev condition for locally finite varieties omitting the unary type. *Algebra Universalis*, 64(1-2):15–20, 2010. ISSN 0002-5240. doi: 10.1007/s00012-010-0082-3.

Walter Taylor. Varieties obeying homotopy laws. *Canadian Journal of Mathematics*, 29(3):498–527, 1977. ISSN 1496-4279. doi: 10.4153/CJM-1977-054-9.

Dmitriy Zhuk. The proof of CSP dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, 2017. doi: 10.1109/FOCS.2017.38.