Theoretical Computer Science 410 (2009) 1666–1683

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Affine systems of equations and counting infinitary logic

Albert Atserias[a], Andrei Bulatov[b], Anuj Dawar[c,*]

[a] Universitat Politècnica de Catalunya, Barcelona, Spain
[b] Simon Fraser University, Burnaby, Canada
[c] University of Cambridge, Cambridge, UK

ARTICLE INFO

Keywords:
Constraint satisfaction problem
Finite model theory
Universal algebra
Infinitary logic with counting
Tame congruence theory
Bijective game
Systems of linear equations
Inexpressibility
Quantifier-free reductions

ABSTRACT

We study the definability of constraint satisfaction problems (CSPs) in various fixed-point and infinitary logics. We show that testing the solvability of systems of equations over a finite Abelian group, a tractable CSP that was previously known not to be definable in Datalog, is not definable in the infinitary logic with finitely many variables and counting. This implies that it is not definable in least fixed-point logic or its extension with counting. We relate definability of CSPs to their classification obtained from tame congruence theory of the varieties generated by the algebra of polymorphisms of the template structure. In particular, we show that if this variety admits either the unary or affine type, the corresponding CSP is not definable in the infinitary logic with counting.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

The classification of constraint satisfaction problems (CSPs) according to their tractability has been a major research goal since Feder and Vardi first formulated their dichotomy conjecture [14]. This classification has been closely linked to logic, with definability in Datalog providing one important uniform explanation for tractability. However, it has long been known that there are tractable CSPs, such as the satisfiability of systems of linear equations over finite fields, which are not definable in Datalog. Bulatov [5] (see also [3]) provides a uniform explanation for the tractability of these by showing that any constraint language that has a Mal'tsev polymorphism is solvable in polynomial time. It has remained an open question, however, whether there is an explanation for the tractability of these CSPs in terms of a natural logic whose data complexity is in polynomial time and which can define these problems.

The general form of the *constraint satisfaction problem* takes as instance two finite relational structures **A** and **B** and asks if there is a homomorphism from **A** to **B**. We think of the elements of **A** as the variables of the problem and the universe of **B** as the domain of values which these variables may take. The individual tuples in the relations of **A** act as constraints on the values that must be matched to the relations holding in **B**. The general form of the problem is **NP**-complete [26, 27]. In this paper we are mainly concerned with the non-uniform version of the problem which gives rise, for each fixed finite structure **B** to a different decision problem that we denote CSP(**B**), namely the problem of deciding whether a given **A** maps homomorphically to **B**. For many fixed **B**, this problem is solvable in polynomial time, while for others it remains **NP**-complete. A classification of structures for which the problem is tractable remains a major goal of research in the area.

In the present paper we classify constraint satisfaction problems according to their definability in a suitable logic. This is an approach that has proved useful in studying the tractability of constraint satisfaction problems [14,9,23]. In particular, it is known that many natural constraint satisfaction problems that are tractable are definable (or, to be precise, their complements are definable) in Datalog, the language of function-free Horn clauses. Any class of structures that is definable in

[*] Corresponding author.
E-mail address: Anuj.Dawar@cl.cam.ac.uk (A. Dawar).

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.12.049

Datalog is necessarily decidable in polynomial time, but there are known constraint satisfaction problems that are tractable but are not definable in Datalog. A classical example is the solvability of systems of linear equations over the two-element field [14], which we denote CSP($\mathbb{Z}_2$). Furthermore, there are **NP**-complete constraint satisfaction problems, such as 3-colourability of graphs, for which it is possible to show that they are not Datalog-definable without requiring the assumption that **P** is different from **NP**. Indeed, the class of constraint satisfaction problems whose complements are definable in Datalog appears to be a robust, natural class of problems with many independent and equivalent characterisations [10,22].

1.1. Results in logic

A natural question arising from such considerations is whether we can offer any explanation based on logical definability for the tractability of problems such as the satisfiability of systems of linear equations over a finite field. Is there a natural logic such that all problems definable in this logic are polynomial-time decidable and that can express CSP($\mathbb{Z}_2$)? In particular, is this problem definable in LFP – the logic extending first-order logic with least fixed-points or LFP + C – the extension of LFP with counting? These are both logics that have been extensively studied in the context of descriptive complexity as characterising natural fragments of polynomial time. It is a consequence of our results that neither of these logics is able to express the solvability of systems of linear equations over any finite field. Indeed, we show that these problems are not definable in $C^{\omega}_{\infty\omega}$, the infinitary logic with bounded number of variables and counting, a logic much more expressive than LFP + C.

Interestingly, Blass, Gurevich and Shelah [1] proved that LFP + C is able to define the class of non-singular square matrices over any fixed finite field, a result we review in Section 4. Together with our result, this exhibits a fine-grained distinction between the problem of computing the determinant of a square matrix and the problem of computing its rank, as one underlies a problem definable in $C^{\omega}_{\infty\omega}$ and the other underlies a problem that is not. We also note in Section 4 that the problem GAP$_2$ of determining the parity of the number of paths in a graph is also definable in LFP + C. This demonstrates the differences in definability between three natural complete problems for the complexity class $\oplus$**L**, two of which we show are definable in LFP + C but the third is not.

1.2. Results in algebra

Another important means of classifying constraint satisfaction problems is on the basis of the algebra of the template structure **B**. A polymorphism of a structure is an operation of its universe that preserves all its relations (see Section 2 for precise definitions). It is known that whether or not CSP(**B**) is tractable depends only on the algebra $\mathcal{B}$ obtained from the universe of **B** endowed with its polymorphisms. Indeed, it depends only on the variety generated by this algebra. This is established in [4] by showing that if the algebra $\mathcal{B}'$ of structure **B**$'$ is obtained from $\mathcal{B}$ as a power, subalgebra or homomorphic image, then CSP(**B**$'$) is polynomial-time reducible to CSP(**B**). We show in the present paper that this can be improved to Datalog-definable reductions. These are weak reductions that, in particular, preserve definability in LFP and $C^{\omega}_{\infty\omega}$. This allows us to establish that definability of a CSP in these logics is also determined by var($\mathcal{B}$), the variety generated by the algebra of **B**.

Using the tool of Datalog-reductions, which we expect to be useful for other applications in the area, we relate definability of constraint satisfaction problems in $C^{\omega}_{\infty\omega}$ to the classification of varieties of finite algebras from tame congruence theory [18]. It is known [4] that CSP(**B**) is **NP**-complete if var($\mathcal{B}$) admits the unary type (also known as type **1**), and it is conjectured that CSP(**B**) is in **P** otherwise. Similarly, Larose and Zadori showed [24] that CSP(**B**) is not definable in Datalog if var($\mathcal{B}$) admits the unary or affine types (types **1** and **2**), and conjectured the converse. It is a consequence of our results that we can strengthen the assertion by replacing Datalog with $C^{\omega}_{\infty\omega}$. This implies that, if the Larose–Zadori conjecture is true, we obtain a dichotomy of definability whereby, for every **B**, either CSP(**B**) is definable in Datalog or it is not definable in $C^{\omega}_{\infty\omega}$.

1.3. Organization of the paper

The rest of the paper is structured as follows. In Section 2 we present some background definitions. Section 3 gives a proof that solvability of linear equations is not definable in $C^{\omega}_{\infty\omega}$. Section 4 considers the definability of three $\oplus$**L**-complete problems in $C^{\omega}_{\infty\omega}$. Section 5 establishes that the definability of CSP(**B**) is determined by the variety generated by the algebra of **B**. Section 6 begins by showing that if the variety admits the unary or affine type, then it contains an algebra with the operations of a module. These results are tied together in the same section to obtain the main conclusion relating definability in $C^{\omega}_{\infty\omega}$ to the omitting of types from tame congruence theory.

2. Preliminaries

2.1. Notation, relational structures, and homomorphisms

We use the boldface notation **a** for a tuple with components $(a_1, \ldots, a_r)$, and similarly for other letters **b**, **x**, etc. and variants with subindices or primes. The arity of the tuple will be clear from context. If I is a sequence $(i_1, \ldots, i_m)$ of indices in $\{1, \dots, r\}$, we write $\mathbf{a}_I$ for the tuple $(a_{i_1}, \dots, a_{i_m})$. Sometimes we also write $(a_i : i \in I)$ if the meaning is clear. If R is a set of tuples of arity r, we write $\mathrm{pr}_I(R)$ for $\{\mathbf{a}_I : \mathbf{a} \in R\}$.

A vocabulary σ is a finite collection of relation symbols, each with an associated arity. A σ-structure $\mathbf{A}$ consists of a finite set A with a relation $R^{\mathbf{A}} \subseteq A^r$ for each r-ary relation symbol R in σ. A graph is a structure with a binary relation that is symmetric and irreflexive. A *homomorphism* from a σ-structure $\mathbf{A}$ to a σ-structure $\mathbf{B}$ is a map $h : A \to B$ such that for each R in σ and each $(a_1, \dots, a_r) \in A^r$,

$$\text{if } (a_1, \dots, a_r) \in R^{\mathbf{A}}, \text{ then } (h(a_1), \dots, h(a_r)) \in R^{\mathbf{B}}.$$

We write $\mathbf{A} \to \mathbf{B}$ to denote that there exists a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. A structure $\mathbf{A}$ is a *core* if every homomorphism from $\mathbf{A}$ to itself is an automorphism. We write CSP($\mathbf{B}$) for the class of finite structures $\mathbf{A}$ such that $\mathbf{A} \to \mathbf{B}$ and also for the decision problem of determining membership in this class.

2.2. Logic

We assume familiarity with first-order logic. A formula is *quantifier-free* if it has no quantifiers, and *positive quantifier-free* if it is quantifier-free and it has no negations. A formula is *positive primitive* if it is formed from the atomic formulas using only conjunctions and existential quantification. A formula is *existential positive* if it is formed from the atomic formulas using conjunctions, disjunctions and existential quantification. Datalog can be seen as the extension of existential positive formulas with a recursion mechanism for building least fixed-points. Similarly, LFP is the extension of first-order logic with an operator for forming the least fixed-points of positive formulas. Finally, LFP + C is the extension of LFP with a counting mechanism. For formal definitions, which we will not need in this paper, we refer the reader to [25]. It is known that every class of structures definable in LFP + C is decidable in polynomial time.

The formulas of the logic $\mathrm{C}_{\infty\omega}$ are obtained from the atomic formulas using negation, infinitary conjunction and disjunction, and counting quantifiers $\exists^{\geq i}x$ for any integer $i \geq 0$). The fragment $\mathrm{C}^k_{\infty\omega}$ consists of those formulas of $\mathrm{C}_{\infty\omega}$ in which only k distinct variables appear and

$$\mathrm{C}^{\omega}_{\infty\omega} = \bigcup_{k\in\omega} \mathrm{C}^k_{\infty\omega}.$$

The significance of $\mathrm{C}^{\omega}_{\infty\omega}$ is that fixed-point logics can be translated into it. That is, any formula of Datalog or LFP, and indeed of LFP + C is equivalent to one of $\mathrm{C}^{\omega}_{\infty\omega}$. Thus, these logics are fragments of $\mathrm{C}^{\omega}_{\infty\omega}$. Moreover, these translations into infinitary logics have provided some of the most effective tools for proving inexpressibility results for the fixed-point logics. See [13, 19] for a discussion of this and the role of these logics in descriptive complexity.

2.3. Logical reducibilities

Let σ and τ be two finite vocabularies. Let $R_1, \dots, R_s$ be the relation symbols of τ, with arities $r_1, \dots, r_s$. A *k-ary interpretation with p parameters* (of τ in σ) is an $(s+1)$-tuple

$$\mathbf{I} = (\varphi_U, \varphi_1, \dots, \varphi_s)$$

of formulas over the vocabulary σ. The formula $\varphi_U = \varphi_U(\mathbf{x}, \mathbf{y})$ has $k + p$ free variables $\mathbf{x}$ and $\mathbf{y}$. Each formula $\varphi_i = \varphi_i(\mathbf{x}_1, \dots, \mathbf{x}_r, \mathbf{y})$ has $kr_i + p$ free variables $\mathbf{x}_1, \dots, \mathbf{x}_r$ and $\mathbf{y}$. If each formula in $\mathbf{I}$ belongs to a class of formulas Θ, we say that $\mathbf{I}$ is a Θ-interpretation.

Let $\mathbf{A}$ be a σ-structure with universe A and let $\mathbf{c}$ be a p-tuple of points in A. The *image of* $\mathbf{A}$ *through* $\mathbf{I}$ *with parameters* $\mathbf{c}$, denoted by $\mathbf{I}(\mathbf{A}, \mathbf{c})$, is the τ-structure whose universe U is the set of tuples $\mathbf{a}$ in A^k such that $\mathbf{A} \models \varphi_U(\mathbf{a}, \mathbf{c})$, and whose interpretation for R_i is the set of tuples $(\mathbf{a}_1, \dots, \mathbf{a}_r)$ in U^r such that $\mathbf{A} \models \varphi_i(\mathbf{a}_1, \dots, \mathbf{a}_r, \mathbf{c})$. Now we are ready to define the notion of logical reduction:

Definition 1. Let $\mathcal{C}$ and $\mathcal{D}$ be classes of structures and let Θ be a class of formulas. We say that $\mathcal{C}$ reduces to $\mathcal{D}$ under Θ-reducibility, denoted by $\mathcal{C} \leq_{\Theta} \mathcal{D}$, if there exists a Θ-interpretation $\mathbf{I}$ such that, for every structure $\mathbf{A}$ with at least p points, where p is the number of parameters of $\mathbf{I}$, the following are equivalent:

(1) $\mathbf{A} \in \mathcal{C}$
(2) $\mathbf{I}(\mathbf{A}, \mathbf{c}) \in \mathcal{D}$ for every proper $\mathbf{c}$,
(3) $\mathbf{I}(\mathbf{A}, \mathbf{c}) \in \mathcal{D}$ for some proper $\mathbf{c}$,

where a proper $\mathbf{c}$ is a tuple $(c_1, \dots, c_p)$ of points in $\mathbf{A}$ such that $c_i \neq c_j$ whenever $i \neq j$.

We will use for Θ the collections of positive quantifier-free formulas, existential positive formulas, and Datalog formulas (i.e. Datalog programs) and write $\leq_{\mathrm{pqf}}$, $\leq_{\mathrm{ep}}$ and $\leq_{\mathrm{datalog}}$, respectively. Note that these are reducibilities of increasing power, and that definability in $\mathrm{C}^{\omega}_{\infty\omega}$ is preserved downwards by all three (the finitely many exceptions of structures with less than p points can be handled individually). Also, all three reducibilities are transitive.

2.4. Universal algebra

An n-ary operation f on a set A is a *polymorphism* of a relation $R \subseteq A^r$ if, for any tuples $\mathbf{a}_1, \dots, \mathbf{a}_n \in R$, the tuple $f(\mathbf{a}_1, \dots, \mathbf{a}_n)$ obtained by applying f component-wise also belongs to R. We say that R is *invariant* under f. The set of all polymorphisms of a collection of relations F is denoted by $\mathrm{Pol}(F)$, and the set of all invariant relations of a collection of operations C is denoted by $\mathrm{Inv}(C)$. For a relational structure $\mathbf{A}$, we use $\mathrm{Pol}(\mathbf{A})$ for the set of operations on A that are polymorphisms of every relation of $\mathbf{A}$. The following theorem links polymorphisms and definability of relations by positive primitive formulas (pp-formulas).

Theorem 2 ([15,2]). *Let* $\mathbf{A}$ *be a finite structure, and let* $R \subseteq A^r$ *be a non-empty relation that is preserved by all polymorphisms of* $\mathbf{A}$*. Then* R *is definable in* $\mathbf{A}$ *by a pp-formula.*

In [20,21], Jeavons et al. proved that the set of polymorphisms of $\mathbf{B}$ is included in the set of polymorphisms of $\mathbf{A}$, then $\mathrm{CSP}(\mathbf{A})$ is reducible to $\mathrm{CSP}(\mathbf{B})$ by polynomial-time many-one reducibility. Using the recent logarithmic space algorithm for undirected graph reachability [28], the reduction can be made logspace.

Theorem 3 ([20,21]). *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be finite structures. If* $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$*, then* $\mathrm{CSP}(\mathbf{B}) \leq_m^{\log} \mathrm{CSP}(\mathbf{A})$.

A set with a collection of operations on it is called an *algebra*. Two algebras are *term-equivalent* if the sets of operations obtained by composition from the basic operations of the algebra and all the projections are the same in both algebras. As is common in universal algebra, we identify algebras up to term-equivalence.

Every structure $\mathbf{A}$ can be naturally associated with an algebra $\mathrm{Alg}(\mathbf{A})$, called the *algebra* of $\mathbf{A}$, whose base set is the universe of $\mathbf{A}$, and whose operations are the polymorphisms of $\mathbf{A}$. Let $\mathcal{A} = (A, C)$ and $\mathcal{A}' = (A', C')$ be algebras. We say that $\mathcal{A}$ and $\mathcal{A}'$ are *similar*, or of the *same type*, if there exists an index set I such that $C = \{f_i \mid i \in I\}$ and $C' = \{f_i' \mid i \in I\}$, and the operations f_i and f_i' are of the same arity, say n_i, for every $i \in I$. A *homomorphism* from $\mathcal{A}$ to $\mathcal{A}'$ is a map $\varphi : A \to A'$ such that for every $i \in I$ and $a_1, \dots, a_{n_i} \in A$, it holds that $\varphi(f_i(a_1, \dots, a_{n_i})) = f_i'(\varphi(a_1), \dots, \varphi(a_{n_i}))$.

We shall use the four standard ways of transforming algebras.

(1) $\mathcal{A}'$ is a *reduct* of $\mathcal{A}$ if $A' = A$ and $C' \subseteq C$;
(2) $\mathcal{A}'$ is a *subalgebra* of $\mathcal{A}$ if $A' \subseteq A$, every operation from C is a polymorphism of A' treated as a unary relation on A, and C' consists of the operations from C restricted to A'.
(3) $\mathcal{A}'$ is a *homomorphic image* of $\mathcal{A}$ if there exists a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$ that is onto.
(4) $\mathcal{A}'$ is a *direct power* of $\mathcal{A}$ if there exists an integer $k \geq 0$ such that $A' = A^k$ and C' consists of the operations from C acting component-wise on A^k. We write $\mathcal{A}' = \mathcal{A}^k$.

A *variety* is a class of algebras which, if it contains $\mathcal{A}$ also contains every subalgebra of $\mathcal{A}$, every homomorphic image of $\mathcal{A}$, and every direct power of $\mathcal{A}$. The smallest variety containing $\mathcal{A}$ is called the *variety generated* by $\mathcal{A}$ and is denoted by $\mathrm{var}(\mathcal{A})$. For further background on universal algebra, see [8]. We shall have occasion to use the following simple observation on pp-definability and reducts.

Observation 4. Let $\mathbf{A}$ and $\mathbf{B}$ be finite structures with the same universe. The algebra $\mathrm{Alg}(\mathbf{A})$ is a reduct of $\mathrm{Alg}(\mathbf{B})$ if, and only if, every relation of $\mathbf{B}$ is pp-definable in $\mathbf{A}$.

The following theorem uses the above mentioned result by Jeavons et al. and the results of [4].

Theorem 5. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be finite structures. If the variety generated by* $\mathrm{Alg}(\mathbf{A})$ *contains a reduct of* $\mathrm{Alg}(\mathbf{B})$ *then* $\mathrm{CSP}(\mathbf{B}) \leq_m^{\log} \mathrm{CSP}(\mathbf{A})$.

Note that Theorem 3 is a direct consequence of this because $\mathrm{Alg}(\mathbf{A})$ belongs to the variety it generates, and $\mathrm{Alg}(\mathbf{A})$ is a reduct of $\mathrm{Alg}(\mathbf{B})$ precisely when $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$.

3. Definability and systems of equations

In this section we show that the problem of determining the solvability of linear equations over the two-element field, which we mentioned above as a canonical example of a tractable CSP whose complement is not definable in Datalog, is also not definable in $\mathrm{C}^{\omega}_{\infty\omega}$. Indeed, we prove a more general result by showing that the solvability of equations over a finite Abelian group with at least two elements is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$.

3.1. Combinatorial games

Our proof of undefinability is based on a game argument. The expressive power of $C^{\omega}_{\infty\omega}$ is characterised by a game known as the *bijective game* [16]. This is played by two players, Spoiler and Duplicator, on a pair of structures **A** and **B**, with k pairs of pebbles (x_i, y_i) for $1 \leq i \leq k$. For each move, Spoiler chooses a pair of pebbles (x_i, y_i), Duplicator chooses a bijection $f : A \to B$ such that $f(x_j) = y_j$ for $i \neq j$, and Spoiler chooses $a \in A$ and places x_i on a and y_i on $f(a)$. If, after some move, the map $\mathbf{x} \mapsto \mathbf{y}$ is not a partial isomorphism, Spoiler wins; Duplicator wins infinite plays. By a result of Hella [16], Duplicator has a winning strategy if, and only if, **A** and **B** cannot be distinguished by any formula of $C^{k}_{\infty\omega}$, a fact denoted by $\mathbf{A} \equiv^{\mathcal{C}^k} \mathbf{B}$.

In constructing the winning strategy in the bijective game we construct, we depend on another game, the *cops and robber game* [29], which is known to characterise the treewidth of a graph. For the standard definition of the treewidth of a graph, we refer the reader to [12]. The cops and robber game is played by two players, one of whom controls the set of k cops attempting to catch a robber controlled by the other player. The cop player can move any set of cops to any vertices of the graph, while the robber can move along any path in the graph as long as there is no cop currently on the path. It is known [29] that the cop player has a winning strategy on a graph using $k + 1$ cops if and only if the graph has treewidth at most k. The treewidth of a graph G is denoted $tw(G)$.

3.2. Systems of equations as a CSP

We now turn to the precise formulation of the problem of deciding the solvability of equations over a finite Abelian group $\mathcal{G}$ as a class of relational structures. In the following we will write $+$ for the group operation in $\mathcal{G}$ and 0 for the identity.

Definition 6. Let $\mathcal{G}$ be a finite Abelian group over a set G and r be a positive integer. We define the structure $\mathbf{E}_{\mathcal{G},r}$ to have universe G and, for each $a \in G$ and $1 \leq j \leq r$, it has a relation R^j_a of arity j that consists of the set of tuples $(x_1, \dots, x_j) \in G^j$ that satisfy the equation $x_1 + \cdots + x_j = a$.

Thus, any structure **A** in the signature of $\mathbf{E}_{\mathcal{G},r}$ can be seen as a set of equations in which at most r variables occur in each equation. The universe of **A** is the set of variables and the occurrence of a tuple $(x_1, \dots, x_j)$ in a relation R^j_a signifies the equation $x_1 + \cdots + x_j = a$. This set of equations is solvable if, and only if, $\mathbf{A} \to \mathbf{E}_{\mathcal{G},r}$. In the sequel we will say "the equation $x_1 + \cdots + x_j = a$ occurs in **A**" to mean that the tuple $(x_1, \dots, x_j)$ is in R^j_a.

Our aim now is to exhibit, for each non-trivial finite Abelian group $\mathcal{G}$ and each positive integer k, a pair of structures **A** and **B** such that $\mathbf{A} \equiv^{\mathcal{C}^k} \mathbf{B}$ and such that $\mathbf{A} \in \text{CSP}(\mathbf{E}_{\mathcal{G},3})$ and $\mathbf{B} \notin \text{CSP}(\mathbf{E}_{\mathcal{G},3})$. This will show that $\text{CSP}(\mathbf{E}_{\mathcal{G},3})$ is not definable in $C^{\omega}_{\infty\omega}$. This, of course, implies the result for all $\text{CSP}(\mathbf{E}_{\mathcal{G},r})$ with $r \geq 3$.

The structures we construct are sets of equations derived from 3-regular graphs of large treewidth. From now on, fix a non-trivial finite Abelian group $\mathcal{G}$, a 3-regular graph H, and a distinguished vertex u of H. Let $\{a_1, \dots, a_m\}$ be the elements of $\mathcal{G}$. We define, for each $a \in \{a_1, \dots, a_m\}$, a set of equations $\mathbf{E}_a H^u$ as follows (note that $\mathbf{E}_a H^u$ is a structure over the vocabulary of $\mathbf{E}_{\mathcal{G},3}$):

For each vertex $v \in V^H$ and each edge $e \in E^H$ that is incident on v, we have m distinct variables $x^{v,e}_i$ where i ranges over $\{a_1, \dots, a_m\}$. Since each vertex has three edges incident on it, there are $3m$ variables associated to each vertex. For every vertex v *other than* u, let e_1, e_2, e_3 be the three edges incident on v. We then include the following equation in $\mathbf{E}_a H^u$ for all $i, j, k \in \{a_1, \dots, a_m\}$:

$$x^{v,e_1}_i + x^{v,e_2}_j + x^{v,e_3}_k = i + j + k. \tag{1}$$

For the distinguished vertex u, instead of the above, we include the following equation, again for all $i, j, k \in \{a_1, \dots, a_m\}$:

$$x^{u,e_1}_i + x^{u,e_2}_j + x^{u,e_3}_k = i + j + k + a. \tag{2}$$

In addition, for each edge $e \in E^H$ let v_1, v_2 be its endpoints. We include the following equations in $\mathbf{E}_a H^u$ for all $i, j \in \{a_1, \dots, a_m\}$:

$$x^{v_1,e}_i + x^{v_2,e}_j = i + j. \tag{3}$$

We refer to equations of the form (1) and (2) as *vertex equations* and equations of the form (3) as *edge equations*.

Lemma 7. *$\mathbf{E}_a H^u$ is satisfiable if, and only if, $a = 0$.*

Proof. To see that the system of equations $\mathbf{E}_0 H^u$ is satisfiable, just take the assignment that gives the variable $x^{v,e}_i$ the value i.

To see that $\mathbf{E}_a H^u$ is unsatisfiable when $a \neq 0$, consider the subsystem S_0 of equations involving only the variables $x^{v,e}_0$ with subscript 0. Note that each such variable occurs exactly twice in S_0, once in a vertex equation and once in an edge equation. Thus, if we add up the left-hand sides of all the equations, we get $2 \sum x^{v,e}_0$. Note also that each variable $x^{v,e}_0$ has a companion variable $x^{v',e}_0$ where v' is the other endpoint of the edge e and we have the equation $x^{v,e}_0 + x^{v',e}_0 = 0$. Thus

$$2 \sum_{v,e} x^{v,e}_0 = 2 \sum_{e} (x^{v,e}_0 + x^{v',e}_0) = 0.$$

On the other hand, the right-hand side of all equations is 0 except for the one vertex equation for u, which has right-hand side a. Thus summing the right-hand sides of all equations gives the sum a. Since $a \neq 0$, this shows that the subsystem S_0 and hence the system of equations $\mathbf{E}_a H^u$ is unsatisfiable. □

3.3. Winning strategy

Next we argue that any two systems defined this way are sufficiently indistinguishable. We start by showing that the shape of the system does not depend on which distinguished vertex we pick, provided we pick them in the same connected component.

Lemma 8. *If $u, u' \in V^H$ belong to the same connected component of H, then $\mathbf{E}_a H^u \cong \mathbf{E}_a H^{u'}$.*

Proof. The case where $u = u'$ is trivial, so assume that they are distinct.

Let $u = v_1, e_1, \ldots, e_s, v_{s+1} = u'$ be the sequence of vertices and edges along a simple path from u to u'. We now define a map η from $\mathbf{E}_a H^u$ to $\mathbf{E}_a H^{u'}$ as follows:

- for any $v \notin \{v_1, \ldots, v_{s+1}\}$, $\eta(x_j^{v,e}) = x_j^{v,e}$;
- for each $l \in \{1, \ldots, s\}$, $\eta(x_j^{v_l,e_l}) = x_{j+a}^{v_l,e_l}$; and
- for each $l \in \{1, \ldots, s\}$, $\eta(x_j^{v_{l+1},e_l}) = x_{j-a}^{v_{l+1},e_l}$.

To show that η is an isomorphism, we need to argue that it preserves all the equations in $\mathbf{E}_a H^u$. Clearly, all equations corresponding to vertices and edges of H that do not appear on the path are preserved as η is the identity map on the corresponding variables. Consider now the vertex equations corresponding to the vertex u. Note that the edge e_1 (the first edge on the chosen path) is incident on u and let f and g be the two other edges incident on u. Then, the equation

$$x_i^{u,e_1} + x_j^{u,f} + x_k^{u,g} = i + j + k + a$$

is mapped by η to

$$x_{i+a}^{u,e_1} + x_j^{u,f} + x_k^{u,g} = i + j + k + a$$

which is, indeed, an equation of $\mathbf{E}_r H^{u'}$.

Similarly, a vertex equation for u':

$$x_i^{u',e_s} + x_j^{u',f} + x_k^{u',g} = i + j + k$$

is mapped to

$$x_{i-a}^{u',e_s} + x_j^{u,f} + x_k^{u,g} = i + j + k.$$

Now, consider a vertex equation for an intermediate vertex $v = v_{l+1}$ along the path. In this case, there are two edges e_l, e_{l+1} of the path incident on v. Thus, the equation

$$x_i^{v,e_l} + x_j^{v,e_{l+1}} + x_k^{v,f} = i + j + k$$

is mapped by η to

$$x_{i-a}^{v,e_l} + x_{j+a}^{v,e_{l+1}} + x_k^{v,f} = i + j + k,$$

where f is the third edge incident on v.

Finally, for each edge e_l along the path, the equation

$$x_i^{v_l,e_l} + x_j^{v_{l+1},e_l} = i + j$$

is mapped by η to

$$x_{i+a}^{v_l,e_l} + x_{j-a}^{v_2,e} = i + j.$$

We have thus established that η maps equations to equations. Since η is a bijection, and the number of equations in $\mathbf{E}_a H^u$ and in $\mathbf{E}_a H^{u'}$ is the same, this proves that it is an isomorphism. □

Lemma 9. *If $\mathrm{tw}(H) > k$ and H is connected, then $\mathbf{E}_0 H^u \equiv^{\mathcal{C}^k} \mathbf{E}_a H^u$ for any $a \in \mathcal{G}$.*

Proof. Our aim is to exhibit a winning strategy for Duplicator in the k-pebble bijective game played on the two structures $\mathbf{A} = \mathbf{E}_0 H^u$ and $\mathbf{B} = \mathbf{E}_a H^u$. Since $\mathrm{tw}(H) > k$, we know that in the k cops and robber game played on H, robber has a winning strategy and we show how Duplicator can make use of this strategy.

For each vertex $v \in V^H$ let X^v denote the set of variables $x_i^{v,e}$ for edges e incident on v. Similarly, for each $e \in E^H$, let X^e denote the set of variables involving e.

We say that a bijection $f : \mathbf{A} \to \mathbf{B}$ is *good* for a vertex $v \in V^H$ if the following conditions hold:

(1) for all $w \in V^H, fX^w = X^w$;
(2) for all $e \in E^H, fX^e = X^e$;
(3) for all x, y, if $x + y = i$ is an equation in $\mathbf{A}$ then $f(x) + f(y) = i$ is an equation in $\mathbf{B}$; and

(4) for all x, y, z, if $x + y + z = i$ is an equation in $\mathbf{A}$, then
- $f(x) + f(y) + f(z) = i$ is an equation in $\mathbf{B}$ if $x, y, z \notin X^v$; and
- $f(x) + f(y) + f(z) = i + a$ is an equation in $\mathbf{B}$ if $x, y, z \in X^v$.

Observe that the identity is a bijection that is good for u. Also, observe that a bijection that is good for v preserves all equations except the vertex equations for v.

Claim 1. *Given a bijection $f : \mathbf{A} \to \mathbf{B}$ that is good for v, if there is a path in H from v to w avoiding $u_1, \dots, u_k$ then there is a bijection $f' : \mathbf{A} \to \mathbf{B}$ that is good for w such that $f|_{(X^{u_1} \cup \dots \cup X^{u_k})} = f'|_{(X^{u_1} \cup \dots \cup X^{u_k})}$.*

Proof. Let the path from v to w avoiding $u_1, \dots, u_k$ be $v = v_1, \dots, v_n = w$. For each edge $e = \{v_i, v_{i+1}\}$ along this path, write x_j^{e-} for the variable $x_j^{v_i,e}$ and x_j^{e+} for the variable $x_j^{v_{i+1},e}$. We then define f' by $f'(x_j^{e-}) = f(x_{j-a}^{e-})$ and $f'(x_j^{e+}) = f(x_{j+a}^{e+})$; and f' agrees with f everywhere else. In particular, since the path from v to w avoids $u_1, \dots, u_k$, f' agrees with f on $X^{u_1} \cup \dots \cup X^{u_k}$. □

We now describe Duplicator's winning strategy in the bijective k-pebble game. Duplicator responds to Spoiler's first move with the identity bijection. She maintains a board on the side which describes a position in the k cops and robber game played on the graph H. At any point in the game, if Spoiler's pebbles are on the position $x_1, \dots, x_k$ in $\mathbf{A}$ and $v_1, \dots, v_k$ are the vertices of H to which these variables correspond, then the current position of the cops and robber game has k cops sitting on the vertices $v_1, \dots, v_k$. If the robber's position according to its winning strategy is v, then Duplicator will play a bijection that is good for v.

To see that Duplicator can do this forever, suppose Spoiler lifts a pebble from x_i. Duplicator responds with a current bijection f that is good for v. Since the only equations not preserved by f are those associated with the vertex v, Spoiler must place at least three pebbles on variables associated with v to win the game. However, Duplicator responds to Spoiler placing the pebble on a new position x_i' by updating the position of the cops and robber game. Suppose robber's winning strategy dictates that the robber move from v to w. Since robber's move must be along a path avoiding the current cop positions, by Claim 1, Duplicator can update the bijection f to a new f' that is good for w without changing f on any of the currently pebbled positions. It is now clear that Duplicator can play forever. □

3.4. Undefinability result

We are ready to state and prove the main result of this section:

Theorem 10. *Let $\mathcal{G}$ be a non-trivial finite Abelian group. Then $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$.*

Proof. Suppose, to the contrary, that there is a k such that $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ is definable in $\mathrm{C}^{k}_{\infty\omega}$. Let H be any connected, 3-regular graph with $\mathrm{tw}(H) > k$ and u any vertex of H. For instance, H could be a sufficiently large brick graph. Let a be any element of $\mathcal{G}$ distinct from 0. Then, by Lemma 7, $\mathbf{E}_0 H^u \in \mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ and $\mathbf{E}_a H^u \notin \mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$. But, by Lemma 9, $\mathbf{E}_0 H^u \equiv^{\mathcal{C}^k} \mathbf{E}_a H^u$, a contradiction. □

4. Expressive power of counting logics

The results of Section 3 show that the counting logic $\mathrm{C}^{\omega}_{\infty\omega}$ cannot express the satisfiability of systems of linear equations over a finite field. Over the two-element field $\mathbb{Z}_2$, this problem is complete for the complexity class $\oplus\mathbf{L}$, for which we give a definition in Section 4.4. As we noted in Section 1, however, $\mathrm{C}^{\omega}_{\infty\omega}$ and even its uniform fragment LFP + C are powerful enough to express other complete problems for this class. For example, LFP + C can express that a given square matrix has non-zero determinant over any finite field. This was first noted by Blass, Gurevich and Shelah [1]. In this section, we revisit this result and discuss the (in)expressibility of these and other problems that are complete for $\oplus\mathbf{L}$.

4.1. Counting quantifiers

Recall that the formulas of $\mathrm{C}^{\omega}_{\infty\omega}$ are obtained from the atomic formulas by means of negation, infinitary conjunction and disjunction, and counting quantifiers $(\exists^{\geq n} x)$ for every integer n. The formula $(\exists^{\geq n} x)(\varphi(x))$ signifies that there exist at least n points of the universe that satisfy $\varphi(x)$. In the formulas below we will use the notation

$$(\mathrm{E}\, z)(\varphi(z))$$

as an abbreviation for the infinitary formula

$$\bigvee_{n \geq 0} (\exists^{\geq 2n} z)(\varphi(z)) \wedge \neg(\exists^{\geq 2n+1} z)(\varphi(z))$$

which says that the number of points that satisfy $\varphi(z)$ is even. We also define

$$(\mathrm{O}\, z)(\varphi(z)) \equiv \neg(\mathrm{E}\, z)(\varphi(z))$$

which says that the number of points that satisfy $\varphi(z)$ is odd. As a matter of fact, for every set of integers Q whatsoever, we could write the infinitary formula

$$\bigvee_{n\in Q} (\exists^{\geq n} z)(\varphi(z)) \wedge \neg(\exists^{\geq n+1} z)(\varphi(z))$$

saying that the number of points that satisfy $\varphi(z)$ belongs to Q. Let us mention then, as a curiosity, that an immediate consequence to this is that the infinitary logic $\mathrm{C}^{\omega}_{\infty\omega}$ is able to express non-computable properties of finite structures. Of course, the uniform fragment LFP + C of $\mathrm{C}^{\omega}_{\infty\omega}$ does not have this property as, in fact, every property expressible in LFP + C can be checked in polynomial time.

4.2. Matrix multiplication and powering

An $n \times n$ matrix $A = (a_{ij} : 1 \leq i, j \leq n)$ over the two-element field $\mathbb{Z}_2$, whose elements we denote by 0 and 1, is represented by the binary relation formed by the pairs (i, j) such that $a_{ij} = 1$. In other words, the matrix is represented by the set of positions that hold 1; the other positions hold 0. With some abuse of notation, we will use A both for the matrix in the usual meaning and for the binary relation over $\{1, \dots, n\}$ that represents it.

Products and powers. We define a formula $\mathrm{prod}(x, y, A, B)$ that defines the product of two $n \times n$ matrices A and B over $\mathbb{Z}_2$. This is:

$$\mathrm{prod}(x, y, A, B) \equiv (\mathrm{O}\, z)(A(x, z) \wedge B(z, y)).$$

The particular case in which $A = B$ is denoted by $\mathrm{square}(x, y, A)$.

Next, for every non-negative integer r, we write a formula $\mathrm{power}_r(x, y, A)$ that defines the power A^r of the matrix A. For $r = 0$, the power A^r is simply the identity matrix, which is defined by the formula

$$I(x, y) \equiv (x = y).$$

For $r > 1$, we proceed inductively, so

$$\begin{aligned} \mathrm{power}_0(A) &\equiv I \\ \mathrm{power}_{r+1}(A) &\equiv \mathrm{prod}(A, \mathrm{power}_r(A)). \end{aligned}$$

By carefully reusing variables, it is possible to write the formula $\mathrm{power}_r(x, y, A)$ with four variables in total. For concreteness, we define

$$\begin{aligned} \mathrm{power}_{r+1}(x, y, A) &\equiv (\mathrm{O}\, u)(A(x, u) \wedge \mathrm{power}_r(u, y, A)), \quad \text{if } r \text{ is odd} \\ \mathrm{power}_{r+1}(x, y, A) &\equiv (\mathrm{O}\, v)(A(x, v) \wedge \mathrm{power}_r(v, y, A)), \quad \text{if } r \text{ is even.} \end{aligned}$$

In total, we used four variables x, y, u and v.

Repeated squaring. It is also convenient to define the powering of matrices by a more efficient induction known as *repeated squaring*. For $r = 0$, we use the same base case, and for larger powers we distinguish the odd and even cases:

$$\begin{aligned} \mathrm{power}'_0(A) &\equiv I \\ \mathrm{power}'_{r+1}(A) &\equiv \mathrm{prod}(A, \mathrm{square}(\mathrm{power}'_{\lfloor (r+1)/2 \rfloor}(A))), \quad \text{if } r \text{ is even} \\ \mathrm{power}'_{r+1}(A) &\equiv \mathrm{prod}(I, \mathrm{square}(\mathrm{power}'_{\lfloor (r+1)/2 \rfloor}(A))), \quad \text{if } r \text{ is odd.} \end{aligned}$$

Note that, by the same careful reusing of variables as we did for power_r, it is possible to write the formula power'_r with five variables (the additional variable is for the O-quantifier in square).

The good feature of the inductive definition based on repeated squaring is that it takes only $O(\log r)$ iterations to obtain power'_r. In comparison, the inductive definition of power_r requires r iterations. This difference is important if we need to take powers that are exponential in n, the dimension of the matrix, as we actually do below.

4.3. Non-singular square matrices

A square matrix is called singular if its determinant is zero, and non-singular otherwise. Equivalently, a square matrix is non-singular if the columns are linearly independent. In the two-element field, a column is a linear combination of some other columns if and only if it is the sum of a subset of those. Therefore, the number of non-singular $n \times n$ matrices over the two-element field is

$$(2^n - 1)(2^n - 2)(2^n - 2^2)(2^n - 2^3) \cdots (2^n - 2^{n-1})$$

because we have $2^n - 1$ choices of non-zero vectors for the first column, and more generally, $2^n - 2^i$ choices of vectors that are not sums of any of the i previous columns for the $(i + 1)$th column. Let N_n be this number.

Non-singular matrices over $\mathbb{Z}_2$. Since the product of non-singular matrices is non-singular and the inverse of a non-singular matrix is non-singular, the collection of non-singular $n \times n$ matrices over the two-element field forms a group of order N_n. Therefore, A is non-singular if and only if $A^{N_n} = A$. It follows that the sentence below expresses that the $n \times n$ matrix A over the two-element field is non-singular:

$$\text{nonsingular}_n(A) \equiv (\forall x)(\forall y)(\text{power}_{N_n}(x, y, A) \leftrightarrow A(x, y)).$$

Finally, if we want to define a sentence nonsingular(A) that defines the class of all finite structures that represent non-singular square matrices A over the two-element field, regardless of its dimensions, it suffices to take

$$\text{nonsingular}(A) \equiv \bigvee_{n \geq 1} (\text{matrix}_n(A) \wedge \text{nonsingular}_{N_n}(A))$$

where matrix$_n(A)$ is the sentence saying that A is a square $n \times n$ matrix. In other words, this is the sentence that says that the structure has exactly n elements:

$$(\exists^{\geq n} x)(x = x) \wedge \neg(\exists^{\geq n+1} x)(x = x).$$

This shows that the collection of non-singular matrices over the two-element field is definable in $C^{\omega}_{\infty\omega}$. Let us mention that all definitions are perfectly uniform and can be formalized in the uniform fragment LFP + C of $C^{\omega}_{\infty\omega}$. For this, it is important to use the inductive definition of power$'_r$ based on repeated squaring because in the definition of non-singular matrices we are taking a power with exponent N_n, which is $2^{O(n^2)}$. Note that the log of this number is polynomial.

Non-singular matrices over finite fields. The discussion up to now can be generalized to arbitrary finite fields. Let F be a finite field with q elements. An $n \times n$ matrix $A = (a_{ij} : 1 \leq i, j \leq n)$ over F is represented by $q - 1$ binary relations R_a, one for each $a \in F - \{0\}$, where R_a is the relation containing the pairs (i, j) such that $a_{ij} = a$. In other words, R_a is the set of positions of the matrix that hold a. The positions that do not belong to any R_a hold 0. It is not too difficult to define $C^{\omega}_{\infty\omega}$ formulas prod(x, y, A, B) and power$_r(x, y, A)$ defining the product and the power of matrices as we did for the two-element field in the previous section. It is important for this that the finite field is fixed.

The set of non-singular matrices over F also forms a group. Its order is

$$(q^n - 1)(q^n - q)(q^n - q^2) \cdots (q^n - q^{n-1})$$

because we have $q^n - 1$ choices for the non-zero vector of the first column, and more generally, $q^n - q^i$ choices of vectors that are not linear combinations of previous columns for the $(i + 1)$th column. If we let $N_{n,q}$ be this number, then an $n \times n$ matrix A over F is a non-singular if and only if $A^{N_{n,q}} = A$. Thus, the collection of all non-singular matrices over F is definable in $C^{\omega}_{\infty\omega}$, and in fact in its uniform fragment LFP + C too.

4.4. Complete problems for $\oplus$**L**

The complexity class $\oplus$**L** is formally defined as follows. It consists of all languages L for which there exists a non-deterministic Turing machine M running in logarithmic space such that, for every input x in L, the number of accepting computations of M is odd, and for every input x not in L, the number of accepting computations of M is even. It can be seen that **NL** $\subseteq \oplus$**L** $\subseteq$ **P**, but neither inclusion is known to be proper. This class was introduced with the aim of classifying important problems of linear algebra [7].

Complete problems. The problem GAP$_2$, for Graph Accessibility Problem mod 2, is this:

> GAP$_2$: Given a directed acyclic graph **G** and two vertices s and t, decide whether the number of paths that go from s to t in **G** is odd.

It is not hard to see, using the standard reductions from logspace Turing machines to graph reachability problems, that GAP$_2$ is complete for $\oplus$**L**. The trick to make the digraph acyclic consists in adding a counter of steps in a separate tape of the logspace machine.

We define two more problems:

> Nonsingular$_2$: Given a matrix A in $\mathbb{Z}_2^{n \times n}$, decide whether A is non-singular.
>
> Feasible$_2$: Given a matrix A in $\mathbb{Z}_2^{m \times n}$ and a vector b in $\mathbb{Z}_2^m$, decide whether the system of equations $Ax = b$ has a solution x in $\mathbb{Z}_2^n$.

It was shown in [7] that the three problems GAP$_2$, Nonsingular$_2$ and Feasible$_2$ are interreducible by reductions that preserve membership in $\oplus$**L** downwards (where $\oplus$**L** is the command for $\oplus$**L**). It follows that all three are complete for $\oplus$**L**. Thus, from the computational complexity perspective, the three problems are equally hard (or easy).

From the descriptive complexity perspective, however, it follows from our results that the situation is different. We showed in this section that the problem Nonsingular$_2$ is definable in $C^{\omega}_{\infty\omega}$. Moreover, it is not hard to see that if A is the adjacency matrix of the digraph **G**, interpreted as a matrix over the two-element field, then the (u, v)-entry of the rth power A^r is the parity of the number of walks of length r that go from u to v in **G**. In other words, if the number of walks of length

r is odd, then the (u, v)-entry is 1, and if it is even, then it is 0. It follows that if A is the binary relation representing the adjacency matrix of a directed acyclic graph **G**, the following formula of $\mathrm{C}^{\omega}_{\infty\omega}$ defines GAP_2:

$$\bigvee_{S\in H}\bigwedge_{r\in S}\mathrm{power}_r(s,t,A),$$

where H is the set of all finite sets of natural numbers of odd cardinality. Here we use the fact that in a directed acyclic graph, the only walks are paths.

For the problem $\textsc{Feasible}_2$ the situation is different. We show below how the results in Section 3 imply that, when appropriately encoded into finite structures, this problem is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$.

Inexpressibility of feasible systems. Let $A = (a_{ij} : 1 \leq i \leq m, 1 \leq j \leq n)$ be a matrix in $\mathbb{Z}_2^{m\times n}$ and let $b = (b_i : 1 \leq i \leq m)$ be a vector in $\mathbb{Z}_2^m$. The system of equations $Ax = b$ is represented by a finite structure as follows. The universe is the disjoint union of two sets $R = \{r_1, \dots, r_m\}$ and $C = \{c_1, \dots, c_n\}$ of sizes m and n, respectively, indexing the rows and columns of A, respectively. The matrix itself is represented by the set of pairs (r_i, c_j) such that $a_{ij} = 1$. Finally, the vector b is represented by the set of r_i's such that $b_i = 1$.

Now we show that the constraint satisfaction problem $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$, where $\mathcal{G}$ is the additive group of $\mathbb{Z}_2$, reduces to $\textsc{Feasible}_2$ by a quantifier-free reduction. Recall that an instance of $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ is given by a finite structure **A** with its universe representing the set of variables $x_1, \dots, x_n$, and with two ternary relations R_0 and R_1 representing equations of the form

$$x_i + x_j + x_k = 0 \quad\text{and}\quad x_i + x_j + x_k = 1,$$

respectively. We build a structure **C** representing an instance $Ax = b$ of $\textsc{Feasible}_2$ as follows. First, if R_0 and R_1 are not disjoint, **C** is just a fixed unsatisfiable instance of $\textsc{Feasible}_2$. Otherwise, the set of columns C of the matrix is $\{x_1, \dots, x_n\}$, the universe of **A** itself. The set of rows R of the matrix is the set of triples in $R_0 \cup R_1$. The union $R \cup C$ is thus the universe of **C**. The binary relation representing the matrix itself is the set of pairs

$$\{((a, b, c), d) \in R \times C : d = a \vee d = b \vee d = c\}.$$

Finally, the subset of R representing the independent vector is precisely R_1. It is obvious that this is a quantifier-free reduction from $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ to $\textsc{Feasible}_2$. Thus, $\textsc{Feasible}_2$ is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$.

5. Logical reductions

The goal of this section is to work out the most useful reductions between CSPs in the framework of logic. Most constructions are more or less standard, but technical, except the reduction to the idempotent case, which requires also a non-trivial twist.

5.1. Expansions by reduced definable relations

Recall that a structure **D** is an expansion of another structure **B** if every relation in **B** is also a relation in **D**. Let us start with the rather trivial case of reductions to expansions.

Lemma 11. *Let* **B** *be a finite structure, and let* **D** *be an expansion of* **B**. *Then,* $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{D})$.

Proof. The transformation that expands every instance of $\mathrm{CSP}(\mathbf{D})$ by empty relations is a reduction from $\mathrm{CSP}(\mathbf{B})$ to $\mathrm{CSP}(\mathbf{D})$. The empty relation is definable by the quantifier-free formula false. □

It should be clear that, in Lemma 11, the converse reduction $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B})$ need not be true. There is an interesting case, however, where it holds. This is when **D** is an expansion of **B** by *reduced* pp-definable relations. Before we prove this, we need a definition.

Let $R \subseteq A^s$ be a relation on A. We define an equivalence relation θ_R on $\{1, \dots, s\}$ by setting (i, j) in θ_R if, and only if, $a_i = a_j$ for every $(a_1, \dots, a_s)$ in R. We say that R is a *reduced* relation if θ_R is the trivial equivalence relation (i.e. equality). Note that the equality relation on A is not reduced.

Lemma 12. *Let* **B** *be a finite structure, and let* **D** *be an expansion of* **B** *by reduced relations that are definable in* **B** *by a pp-formula. Then,* $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B})$.

Proof. Let σ be the vocabulary of **B**. We prove the lemma for the expansion by one reduced relation R. The general case follows by composing. Let r be the arity of R and let $\phi(x_1, \dots, x_r)$ be the primitive-positive formula that defines R in **B**. The formula has the following form:

$$(\exists x_{r+1})\cdots(\exists x_m)\left(R_1(\mathbf{x}_{I^1_1})\wedge\cdots\wedge R_1(\mathbf{x}_{I^1_{n_1}})\wedge\cdots\wedge R_s(\mathbf{x}_{I^s_1})\wedge\cdots\wedge R_s(\mathbf{x}_{I^s_{n_s}})\right),$$

where $R_1, \dots, R_s$ are all the relation symbols of σ, each I^i_j is a sequence of indices in $\{1, \dots, m\}$ whose length matches the arity r_i of R_i, and $\mathbf{x}_I$ denotes the projection of the tuple $(x_1, \dots, x_m)$ to the indices indicated by I. We may assume that all variables $x_{r+1}, \dots, x_m$ are distinct and disjoint from $x_1, \dots, x_r$. Moreover, since R is reduced, we may also assume that all variables $x_1, \dots, x_r$ are distinct. Given an instance $\mathbf{C}$ of CSP($\mathbf{D}$), we need to define an instance $\mathbf{A}$ of CSP($\mathbf{B}$) such that $\mathbf{A} \to \mathbf{B}$ if and only if $\mathbf{C} \to \mathbf{D}$. First we define $\mathbf{A}$ abstractly, and then show how to define it in $\mathbf{C}$ through a positive quantifier-free interpretation with parameters.

The universe of $\mathbf{A}$ is the set

$$C \cup (R^{\mathbf{C}} \times \{x_{r+1}, \dots, x_m\}),$$

where $x_{r+1}, \dots, x_m$ are the quantified variables in ϕ, which we assume not to be members of C. Intuitively, we have a new copy of each quantified variable of ϕ for each tuple in $R^{\mathbf{C}}$. The interpretation of the relation R_i in $\mathbf{A}$ consists of $R_i^{\mathbf{C}}$, together with a set of tuples defined next. For every $\mathbf{c} = (c_1, \dots, c_r)$ in $R^{\mathbf{C}}$ and for every $I_j^i = (i_1, \dots, i_{r_i})$, add to $R_i^{\mathbf{A}}$ the tuple $(z_1, \dots, z_{r_i})$ defined by:

(1) $z_k = c_{i_k}$ if i_k is the index of a free variable of ϕ, that is, $1 \le i_k \le r$,
(2) $z_k = (\mathbf{c}, x_{i_k})$ if i_k is the index of a bound variable of ϕ, that is, $r + 1 \le i_k \le m$.

This defines the structure $\mathbf{A}$. Let us prove it has the right property:

Claim 2. $\mathbf{A} \to \mathbf{B}$ *if and only if* $\mathbf{C} \to \mathbf{D}$.

Proof. Let h be a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. We claim that the restriction of h to C is a homomorphism from $\mathbf{C}$ to $\mathbf{D}$. For every R_i we have $R_i^{\mathbf{C}} \subseteq R_i^{\mathbf{A}}$ and $R_i^{\mathbf{D}} = R_i^{\mathbf{B}}$. Moreover, h is a homomorphism, so $h(R_i^{\mathbf{A}}) \subseteq R_i^{\mathbf{B}}$. Thus $h(R_i^{\mathbf{C}}) \subseteq R_i^{\mathbf{D}}$. Let us now check that $h(R^{\mathbf{C}}) \subseteq R^{\mathbf{D}}$. Then, let $\mathbf{c}$ be any tuple in $R^{\mathbf{C}}$. Let $\mathbf{d} = h(\mathbf{c})$. We want to show that $\mathbf{B} \models \phi(\mathbf{d})$, so $\mathbf{d}$ belongs to $R^{\mathbf{D}}$. By the definition of $\mathbf{A}$, for every $I_j^i = (i_1, \dots, i_{r_i})$, the tuple $(z_1, \dots, z_{r_i})$ defined as before belongs to $R_i^{\mathbf{A}}$. Now, if i_k is the index of a bound variable of ϕ, we view $h((\mathbf{c}, x_{i_k}))$ as a witness for x_{i_k} when evaluating $\phi(\mathbf{d})$ in $\mathbf{B}$. On the other hand, if i_k is the index of a free variable of ϕ, we view $d_{i_k} = h(c_{i_k})$ as the interpretation of x_{i_k}. This interpretation is well-defined because, critically, $R^{\mathbf{D}}$ is reduced so all m variables $x_1, \dots, x_m$ are distinct. Under this interpretation for the free and bound variables, we have $\mathbf{B} \models \phi(\mathbf{d})$ as was to be proved.

Suppose now that h is a homomorphism from $\mathbf{C}$ to $\mathbf{D}$. We need to extend h to map from A to B. Fix a tuple $\mathbf{c}$ in $R^{\mathbf{C}}$, and let $\mathbf{d} = h(\mathbf{c})$. Then $\mathbf{d}$ belongs to $R^{\mathbf{D}}$ so $\mathbf{B} \models \phi(\mathbf{d})$. Let $b_{r+1}, \dots, b_m$ be witnesses to the existentially quantified variables in ϕ. We extend h by defining $h((\mathbf{c}, x_i)) = b_i$ for $r + 1 \le i \le m$. The claim is that h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ and that this follows directly from the definitions. □

We need to show now that this reduction is indeed a positive quantifier-free interpretation with parameters. This is more or less routine. Fix a pair of distinct variables p_0, p_1 that will play the role of parameters. For concreteness, we can think of p_0 and p_1 as distinct elements of C. Let $q = m - r$ and $t = \lfloor \log_2 q \rfloor + 1$. We can think of the universe of $\mathbf{A}$ as the subset of C^{r+t+2} defined by the following formula with $r + t + 2$ free variables $y_0, y_1, \dots, y_{r+t}$:

$$(y_0 = p_0 \wedge y_1 = \cdots = y_{r+t+1}) \vee (y_0 = p_1 \wedge R(y_1, \dots, y_r) \wedge \psi(y_{r+1}, \dots, y_{r+t})),$$

where $\psi(y_{r+1}, \dots, y_{r+t})$ is a formula that is satisfied by the set of numbers $k \in \{0, \dots, q - 1\}$ when encoded in binary; the bits are encoded by $y_{r+b} = p_0$ or $y_{r+b} = p_1$. In other words, when q is an exact power of two, which we may as well assume by adding dummy variables, ψ is the following formula:

$$\bigwedge_{b=0}^{t-1} (y_{r+1+b} = p_0 \vee y_{r+1+b} = p_1).$$

Intuitively, the set of tuples $(y_0, \dots, y_{r+t})$ for which

$$y_0 = p_0 \wedge y_1 = \cdots = y_{r+t}$$

encodes C, and the set of tuples for which

$$y_0 = p_1 \wedge R(y_1, \dots, y_r) \wedge \psi_q(y_{r+1}, \dots, y_{r+t+1})$$

encodes $R^{\mathbf{C}} \times \{x_{r+1}, \dots, x_m\}$. With the universe defined this way, the rest of the formal definition is easy to work out. □

5.2. Expansions by definable relations

The hypothesis in Lemma 12 that all relations expanding $\mathbf{B}$ must be reduced is necessary, if we want to get pqf-reducibilities. However, if we are satisfied with Datalog-reducibilities, we can relax it. Before we prove this, we need a technical intermediate lemma.

Let R be a relation of arity s and recall the definition of θ_R, the equivalence relation on $\{1, \dots, s\}$ defined in the previous section. Let I be a set of representatives of the equivalence-classes of θ_R, ordered in an arbitrary way, and define $\mathrm{red}(R) = \mathrm{pr}_I R$. Note that $\mathrm{red}(R)$ does not depend on the choice of I. Besides, for every $i \notin I$ there exists some $j \in I$ such that $a_i = a_j$ for every tuple $(a_1, \dots, a_s)$ in R. We call $\mathrm{red}(R)$ the reduced version of R.

A *reduced structure* is a structure all whose relations are reduced. To every structure $\mathbf{B}$ we can associate a reduced structure, called the *reduced version of* $\mathbf{B}$, whose universe is the universe of $\mathbf{B}$ itself and whose relations are the reduced versions of the relations of $\mathbf{B}$. Note that the vocabularies of a structure and its reduced version may be different. Note that the polymorphisms of $\mathbf{B}$ and its reduced version are the same.

Lemma 13. *Let* $\mathbf{B}$ *a finite structure and let* $\mathbf{D}$ *be the reduced version of* $\mathbf{B}$*. Then* $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{D})$ *and* $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B})$.

Proof. We start with the reduction $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B})$. Let σ be the vocabulary of $\mathbf{B}$ and let σ' be the vocabulary of the reduced structure $\mathbf{D}$. Hence, for every symbol R in σ we have a symbol R' in σ' of the arity of $\mathrm{red}(R^{\mathbf{B}})$. Let $\mathbf{C}$ be an instance of $\mathrm{CSP}(\mathbf{D})$. We define an instance $\mathbf{A}$ of $\mathrm{CSP}(\mathbf{B})$. The universe of $\mathbf{A}$ is C itself. The interpretation of the r-ary symbol R in $\mathbf{A}$ is defined as follows: let $\theta = \theta_R$ for $R = R^{\mathbf{B}}$ and let I be a set of representatives of the θ-classes, ordered in an arbitrary way. Then, $R^{\mathbf{A}}$ is defined by the formula

$$\psi_R(x_1, \dots, x_r) = R'(\mathbf{x}_I) \wedge \bigwedge_{(i,j)\in\theta} x_i = x_j.$$

It is clear that $\mathbf{C} \to \mathbf{D}$ if, and only if, $\mathbf{A} \to \mathbf{B}$. Moreover, the reduction is positive quantifier-free.

We proceed now with the reduction $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{D})$.

Let $\mathbf{A}$ be an instance of $\mathrm{CSP}(\mathbf{B})$. We define an instance $\mathbf{C}$ of $\mathrm{CSP}(\mathbf{D})$. The universe of $\mathbf{C}$ is A itself. For the relations, the basic idea is to project every relation $R^{\mathbf{A}}$ to the coordinates of a set of representatives I of the θ-classes, where $\theta = \theta_R$. However, before we do that, we need to *close* each $R^{\mathbf{A}}$ under all equalities implied by the equivalences $(i,j) \in \theta$. We do that using Datalog-definable intermediate relations.

So, let E be the binary relation on A defined by the following Datalog program:

$$E(x_i, x_j) :- R(x_1, \dots, x_s)$$
$$E(x, y) :- E(y, x)$$
$$E(x, z) :- E(x, y) \wedge E(y, z),$$

where the first rule is introduced for every symbol R in σ and every $(i,j) \in \theta_{R^{\mathbf{A}}}$. It is obvious that E is an equivalence relation on A; reflexivity follows from the fact that $(i,j) \in \theta_{R^{\mathbf{A}}}$ in the first rule, symmetry is enforced by the second rule, and transitivity is enforced by the third. Next, for every r-ary symbol R in σ, let R' be the relation defined by

$$R'(\mathbf{x}_I) :- R(y_1, \dots, y_s) \wedge E(x_1, y_1) \wedge \cdots \wedge E(x_s, y_s),$$

where I is a set of representatives of the θ-classes ordered in an arbitrary way. This defines $\mathbf{C}$, and we defined it by a Datalog program interpreted on $\mathbf{A}$. It remains to argue that this Datalog-interpretation is indeed a reduction.

Claim 3. *If* h *is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$ *and* $(a, a') \in E$*, then* $h(a) = h(a')$.

Proof. We proceed by induction on the stage on which (a, a') enters the relation E. If it enters in the first stage, then there exist R in σ, $(i,j) \in \theta_{R^{\mathbf{A}}}$, and $\mathbf{a} \in R^{\mathbf{A}}$ such that $a_i = a$ and $a_j = a'$. Since $h(\mathbf{a}) \in R^{\mathbf{A}}$ and $(i,j) \in \theta_R$, it follows that $h(a_i) = h(a_j)$, so $h(a) = h(a')$. The inductive cases follow trivially from symmetry and transitivity of equality. □

Claim 4. $\mathbf{A} \to \mathbf{B}$ *if and only if* $\mathbf{C} \to \mathbf{D}$.

Proof. Suppose that $\mathbf{A} \to \mathbf{B}$ and let h be a homomorphism. We claim that h itself is also a homomorphism from $\mathbf{C}$ to $\mathbf{D}$. Suppose $\mathbf{c} \in R'^{\mathbf{C}}$. Then there exists $\mathbf{a}$ and $\mathbf{a}'$ in $R^{\mathbf{A}}$ such that $\mathbf{a}_I = \mathbf{c}$ and $(a_i, a'_i) \in E$ for every $i \in \{1, \dots, s\}$. Now, $h(\mathbf{a}') \in R^{\mathbf{B}}$ because h is a homomorphism. But also $h(\mathbf{a}) = h(\mathbf{a}')$ by the claim above because $(a_i, a'_i) \in E$ for every i. But then

$$h(\mathbf{c}) = h(\mathbf{a}_I) = h(\mathbf{a})_I = h(\mathbf{a}')_I \in \mathrm{pr}_I(R^{\mathbf{B}}) = \mathrm{red}(R^{\mathbf{B}}) = R'^{\mathbf{D}}.$$

Thus h is a homomorphism from $\mathbf{C}$ to $\mathbf{D}$.

Suppose now that $\mathbf{C} \to \mathbf{D}$ and let h be a homomorphism. For every $a \in A$, let a^E be a fixed representative of the E-equivalence class of a. Let $g(a) = h(a^E)$ for every a. We claim that g is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Suppose $\mathbf{a} \in R^{\mathbf{A}}$. Then $(\mathbf{a}^E)_I \in R'^{\mathbf{C}}$, so $h((\mathbf{a}^E)_I) \in R'^{\mathbf{D}}$. Note that

$$g(\mathbf{a})_I = h(\mathbf{a}^E)_I = h((\mathbf{a}^E)_I) \in R'^{\mathbf{D}} = \mathrm{red}(R^{\mathbf{B}}) = \mathrm{pr}_I(R^{\mathbf{B}}).$$

But then $g(\mathbf{a}) \in R^{\mathbf{B}}$ by the definition of θ_R and I. So g is a homomorphism. □

This completes the proof of the lemma. □

5.3. Reductions through reducts

By combining the preceding lemmas, we obtain the following result which is the analogue of Theorem 3 for logical reducibilities.

Theorem 14. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be finite structures. If* $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$*, then* $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{A})$.

Proof. Let $\mathbf{B}'$ be the reduced version of $\mathbf{B}$. By Lemma 13, we have $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{B}')$. Let $\mathbf{B}''$ be the expansion of $\mathbf{B}'$ with the relations of $\mathbf{A}$. By Lemma 11, we have $\mathrm{CSP}(\mathbf{B}') \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B}'')$. Suppose now that $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$. Note that $\mathbf{B}''$ is an expansion of $\mathbf{A}$ by reduced relations. Moreover, since the polymorphisms of a relation and its reduced version are the same, it follows from $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$ that every relation $R^{\mathbf{B}'}$ of $\mathbf{B}'$ is invariant under every polymorphism of $\mathbf{A}$. Therefore, by Theorem 2 every $R^{\mathbf{B}'}$ is pp-definable in $\mathbf{A}$. By Lemma 12 we have $\mathrm{CSP}(\mathbf{B}') \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{A})$. Composing we get $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{A})$. □

5.4. Powering, subalgebras, and homomorphic images

In this subsection we show how the basic algebraic constructions of powering, subalgebra and homomorphic images can be handled by Datalog-reductions. We start with homomorphic images.

Let **B** be a finite structure and let $\mathcal{B}$ be its corresponding algebra. Suppose $\mathcal{B}'$ is an algebra that has a homomorphic image $\mathcal{A} = h(\mathcal{B}')$ that is a reduct of $\mathcal{B}$. Note that $A = B = h(B')$, i.e. the universes of $\mathcal{A}$ and $\mathcal{B}$ are the same and are the image of the universe of $\mathcal{B}'$ under h. We define a new structure $\mathbf{B}' = \mathrm{pre}(\mathbf{B}, h)$, the *preimage* of **B**, whose universe is B' and whose relations are the preimages $h^{-1}(R^{\mathbf{B}})$ of the relations $R^{\mathbf{B}}$ of **B**.

Lemma 15. *Let the algebras $\mathcal{B}$ and $\mathcal{B}'$, and the structures* **B** *and* $\mathbf{B}' = \mathrm{pre}(\mathbf{B}, h)$ *be as above. Then*

(1) $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B}')$*; and*
(2) $\mathcal{B}'$ *is a reduct of* $\mathrm{Alg}(\mathbf{B}')$.

Proof. (1) We argue that $\mathrm{CSP}(\mathbf{B}) = \mathrm{CSP}(\mathbf{B}')$ by arguing that **B** and $\mathbf{B}'$ are homomorphically equivalent. The homomorphism from $\mathbf{B}'$ to **B** is just h, and this is easy to check. As a homomorphism from **B** to $\mathbf{B}'$ we take any *inverse* of h; that is, any function $f : B \to B'$ such that $f(b)$ belongs to $h^{-1}(b)$ for every $b \in B$. Such a function exists because h is onto B. It is a homomorphism because if $\mathbf{b}$ is a tuple in $R^{\mathbf{B}}$, then $h(f(\mathbf{b})) = \mathbf{b}$, so $f(\mathbf{b}) \in h^{-1}(R^{\mathbf{B}})$.

(2) It suffices to show that every operation of $\mathcal{B}'$ is a polymorphism of $\mathbf{B}'$. Let f' be an m-ary operation of $\mathcal{B}'$, and let f be the corresponding operation of $\mathcal{A}$. Suppose that $\mathbf{a}^1, \ldots, \mathbf{a}^m$ are m tuples that belong to $h^{-1}(R^{\mathbf{B}})$. Then the tuples $h(\mathbf{a}^1), \ldots, h(\mathbf{a}^m)$ all belong to $R^{\mathbf{B}}$. We apply f component-wise and we obtain the tuple

$$(f(h(a_1^1), \ldots, h(a_1^m)), \ldots, f(h(a_r^1), \ldots, h(a_r^m))).$$

Since f is an operation of $\mathcal{A}$, and $\mathcal{A}$ is a reduct of $\mathcal{B}$, it is a polymorphism of **B**, so this tuple belongs to $R^{\mathbf{B}}$. Now, by the choice of f, this tuple is the same as

$$(h(f'(a_1^1, \ldots, a_1^m)), \ldots, h(f'(a_r^1, \ldots, a_r^m))).$$

We conclude that the tuple

$$(f'(a_1^1, \ldots, a_1^m), \ldots, f'(a_r^1, \ldots, a_r^m))$$

belongs to $h^{-1}(R^{\mathbf{B}})$. This proves that f' preserves every relation of $\mathbf{B}'$. □

Let **B** be a finite structure and let $\mathcal{B}$ be its corresponding algebra. Suppose $\mathcal{B}'$ is an algebra that has a subalgebra $\mathcal{A} \subseteq \mathcal{B}'$ that is a reduct of $\mathcal{B}$. Note that $A = B \subseteq B'$, i.e. the universes of $\mathcal{A}$ and $\mathcal{B}$ are the same and are a subset of the universe of $\mathcal{B}'$. We define a new structure $\mathbf{B}' = \mathrm{ext}(\mathbf{B}, B')$, the *extension* of **B**, with universe B' and the same relations as **B**.

Lemma 16. *Let the algebras $\mathcal{B}$ and $\mathcal{B}'$, and the structures* **B** *and* $\mathbf{B}' = \mathrm{ext}(\mathbf{B}, B')$ *be as above. Then*

(1) $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B}')$*; and*
(2) $\mathcal{B}'$ *is a reduct of* $\mathrm{Alg}(\mathbf{B}')$.

Proof. (1) The structures **B** and $\mathbf{B}'$ are homomorphically equivalent. Indeed the identity mapping on B is a homomorphism of **B** to $\mathbf{B}'$, and any mapping $h : B' \to B$ that is the identity on $B \subseteq B'$ and maps elements from $B' \setminus B$ to any element of B is a homomorphism from $\mathbf{B}'$ to **B**.

(2) Let f' be an operation of $\mathcal{B}'$ and let f be the corresponding operation in $\mathcal{A}$. Then f preserves every relation of **B** because $\mathcal{A}$ is a reduct of $\mathcal{B}$. But then, trivially, f' also preserves every relation of $\mathbf{B}'$ because the relations in $\mathbf{B}'$ and **B** are the same. □

Let R be an r-ary relation on the set A^n. Then the *flattening* of R, denoted $\mathrm{fla}(R, n)$, is the rn-ary relation on A that contains all tuples $(x_1, \ldots, x_{rn})$ such that $((x_1, \ldots, x_n), \ldots, (x_{(r-1)n+1}, \ldots, x_{rn})) \in R$. Let **B** be a finite structure and let $\mathcal{B}$ be its corresponding algebra. Suppose $\mathcal{B}'$ is an algebra that has a direct power $\mathcal{A} = \mathcal{B}'^n$ that is a reduct of $\mathcal{B}$. Note that $A = B = B'^n$, i.e. the universes of $\mathcal{A}$ and $\mathcal{B}$ are the same and are the nth power of the universe of $\mathcal{B}'$. We define a new structure $\mathbf{B}' = \mathrm{fla}(\mathbf{B}, n)$, the *flattening* of **B**, whose universe is B and whose relations are the flattenings of the relations of **B**.

Lemma 17. *Let the algebras $\mathcal{B}$ and $\mathcal{B}'$, and the structures* **B** *and* $\mathbf{B}' = \mathrm{fla}(\mathbf{B}, n)$ *be as above. Then*

(1) $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{B}')$*; and*
(2) $\mathcal{B}'$ *is a reduct of* $\mathrm{Alg}(\mathbf{B}')$.

Proof. (1) Given an instance **A** of CSP(**B**), we need to define an instance $\mathbf{A}'$ of $\mathrm{CSP}(\mathbf{B}')$ such that $\mathbf{A} \to \mathbf{B}$ if, and only if, $\mathbf{A}' \to \mathbf{B}'$. First we define $\mathbf{A}'$ abstractly, and then show how to define it on **A** through a positive quantifier-free interpretation with parameters.

The universe of the structure $\mathbf{A}'$ is $A \times \{1, \ldots, n\}$. For every k-ary symbol R in the vocabulary of **B**, we have a corresponding kn-ary symbol R' in the vocabulary of $\mathbf{B}'$. The interpretation of R' in $\mathbf{A}'$ is defined as the set of all tuples

$$((x_1, 1), \ldots, (x_1, n), \ldots, (x_k, 1), \ldots, (x_k, n))$$

such that $(x_1, \ldots, x_k)$ belongs to $R^{\mathbf{A}}$.

First we prove that this structure has the right property. If $\mathbf{A} \to \mathbf{B}$ and h is a homomorphism, then clearly the mapping $h' : A' \to B'$ defined by the condition $h'((x, i)) = h(x)_i$, where $h(x) = (h(x)_1, \ldots, h(x)_n)$, is a homomorphism. Conversely, if h is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$, then the mapping $h'(x) = (h'((x, 1)), \ldots, h'((x, n)))$ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Next we show that this reduction is positive quantifier-free. Fix a pair of distinct variables p_0, p_1 that will play the role of parameters. For concreteness, we can think of p_0 and p_1 as distinct elements of A. Let $t = \lfloor \log_2 n \rfloor + 1$. We can think of the universe of $\mathbf{A}'$ as the subset of A^{t+1} defined by the formula $\psi(y_0, y_1, \ldots, y_t)$ with $t + 1$ free variables that is satisfied by the tuples $(y_0, y_1, \ldots, y_t)$ for which $(y_1, \ldots, y_t)$ encodes a number from $\{0, \ldots, n - 1\}$ in binary; the bits are encoded by $y_b = p_0$ or $y_b = p_1$ for $1 \leq b \leq t$. The interpretation of the relational symbol R' of arity kn is given by the formula

$$\psi_{R'}(\mathbf{y}^1, \ldots, \mathbf{y}^{kn}) \;= R(y_0^1, y_0^{n+1}, \ldots, y_0^{(k-1)n+1}) \wedge \bigwedge_{j=1}^{n} \bigwedge_{i=0}^{k-1} (\mathrm{bin}(y_1^{in+j}, \ldots, y_t^{in+j}) = j - 1),$$

where $\mathrm{bin}(y_1^{in+j}, \ldots, y_t^{in+j}) = j - 1$ abbreviates the expression

$$y_1^{in+j} = b_1 \wedge \cdots \wedge y_t^{in+j} = b_t$$

and $b_1 \ldots b_t$ is the binary representation of $j - 1$.

(2) Since $\mathscr{A}$ is a reduct of $\mathscr{B}$, every relation of $\mathbf{B}$ is invariant with respect to all operations of $\mathscr{A} = \mathscr{B}'^n$. Now it is straightforward that every relation in the flattening of $\mathbf{B}$ is invariant with respect to every operation of $\mathscr{B}'$. □

5.5. Reductions through varieties

Finally, we are ready to state and prove the analogue of Theorem 5 for logical reducibility.

Theorem 18. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be finite structures. If the variety generated by* $\mathrm{Alg}(\mathbf{A})$ *contains a reduct of* $\mathrm{Alg}(\mathbf{B})$, *then* $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{A})$.

Proof. Let $\mathscr{A} = \mathrm{Alg}(\mathbf{A})$ and $\mathscr{B} = \mathrm{Alg}(\mathbf{B})$. Suppose that some algebra $\mathscr{A}'$ of $\mathrm{var}(\mathscr{A})$ is a reduct of $\mathscr{B}$. By the HSP-theorem [8, Theorem 9.5] $\mathscr{A}'$ is a homomorphic image of a subalgebra of a direct power of $\mathscr{A}$. Let $\mathscr{A}_p$, $\mathscr{A}_s$, and $\mathscr{A}_h$ be the direct power, its subalgebra, and the homomorphic image, respectively. We have that $\mathscr{A}' = \mathscr{A}_h$. Let n be such that $\mathscr{A}_p = \mathscr{A}^n$, and let h be a homomorphism from $\mathscr{A}_s$ to $\mathscr{A}_h$.

We use the three intermediate structures

(1) $\mathbf{A}_s = \mathrm{pre}(\mathbf{A}, h)$,
(2) $\mathbf{A}_p = \mathrm{ext}(\mathbf{A}_s, A_p)$,
(3) $\mathbf{A}_f = \mathrm{fla}(\mathbf{A}_p, n)$.

Using the fact that $\mathscr{A}' = \mathscr{A}_h$ is a reduct of $\mathscr{B} = \mathrm{Alg}(\mathbf{B})$, now we apply Lemmas 15–17 in sequence to obtain

(1) $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{A}_s)$ and $\mathscr{A}_s$ is a reduct of $\mathrm{Alg}(\mathbf{A}_s)$,
(2) $\mathrm{CSP}(\mathbf{A}_s) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{A}_p)$ and $\mathscr{A}_p$ is a reduct of $\mathrm{Alg}(\mathbf{A}_p)$,
(3) $\mathrm{CSP}(\mathbf{A}_p) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{A}_f)$, and $\mathscr{A}$ is a reduct of $\mathrm{Alg}(\mathbf{A}_f)$.

The last condition means that $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{A}_f)$. It follows from Theorem 14 that $\mathrm{CSP}(\mathbf{A}_f) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{A})$. Composing, we get $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{A})$. □

5.6. Reduction from the idempotent case

To every finite structure $\mathbf{B}$ we associate a new structure, the *singleton-expansion* of $\mathbf{B}$, by adding one unary relation $\{b\}$ for every $b \in B$. In other words, if $B = \{b_1, \ldots, b_n\}$, then the structure $(\mathbf{B}, \{b_1\}, \ldots, \{b_n\})$ is the singleton-expansion of $\mathbf{B}$. Note that the polymorphisms of the singleton-expansion of $\mathbf{B}$ are exactly the *idempotent* polymorphisms of $\mathbf{B}$, that is polymorphisms f satisfying the identity $f(x, \ldots, x) = x$. Indeed, every singleton set $\{b\}$ is preserved by any idempotent polymorphism of $\mathbf{B}$, and any polymorphism of $\mathbf{B}$ that preserves every singleton set $\{b\}$ must by idempotent.

Lemma 19. *Let* $\mathbf{B}$ *be a finite structure, and let* $\mathbf{D}$ *be the singleton-expansion of* $\mathbf{B}$. *Then* $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{D})$ *and if* $\mathbf{B}$ *is a core with at least two points, then* $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{ep}} \mathrm{CSP}(\mathbf{B})$.

Proof. Since $\mathbf{D}$ is an expansion of $\mathbf{B}$, the reduction $\mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{pqf}} \mathrm{CSP}(\mathbf{D})$ follows from Lemma 11.

Let us now prove that $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{ep}} \mathrm{CSP}(\mathbf{B})$. Given an instance $\mathbf{C}$ of $\mathrm{CSP}(\mathbf{D})$, we need to define an instance $\mathbf{A}$ of $\mathrm{CSP}(\mathbf{B})$ such that $\mathbf{A} \to \mathbf{B}$ if, and only if, $\mathbf{C} \to \mathbf{D}$. First we define $\mathbf{A}$ abstractly, and then show how to define it on $\mathbf{C}$ through an existential positive interpretation with parameters.

The universe of the structure **A** is the disjoint union of C and B. For every relation symbol R of arity r in the vocabulary of **B**, the interpretation of R in **A** is defined by cases: if the sets $P_b^{\mathbf{C}}$ are not pairwise disjoint, we let $R^{\mathbf{A}} = A^r$. Otherwise, we let $R^{\mathbf{A}}$ be the set

$$R^{\mathbf{B}} \cup \bigcup_{u \in F} u(R^{\mathbf{C}}),$$

where F is the set of mappings $u : C \to A$ such that the following two conditions are satisfied:

(1) $u(y) \in P_b^{\mathbf{C}} \cup \{b\}$ for every $b \in B$ and $y \in P_b^{\mathbf{C}}$,
(2) $u(y) = y$ for every $y \in C - \bigcup_{b \in B} P_b^{\mathbf{C}}$.

This defines the structure **A**. Before we show how to define **A** by an existential positive interpretation, let us show that it has the property we want:

Claim 5. $\mathbf{A} \to \mathbf{B}$ *if, and only if,* $\mathbf{C} \to \mathbf{D}$.

Proof. If the sets $P_b^{\mathbf{C}}$ are not pairwise disjoint, then clearly $\mathbf{C} \not\to \mathbf{D}$. In this case, every relation in **A** is the full relation and in particular it is reflexive. But then $\mathbf{A} \not\to \mathbf{B}$ since otherwise **B** would also be reflexive and hence not a core with at least two elements.

Suppose in the following that the sets $P_b^{\mathbf{C}}$ are pairwise disjoint. Let h be a homomorphism from **C** to **D**. Note that $h(y) = b$ for every $y \in P_b^{\mathbf{C}}$; this remark will be of use later. Let g be the unique extension of h to $A = B \cup C$ such that $g(b) = b$ for every $b \in B$. We prove that g is a homomorphism from **A** to **B**. Let $\mathbf{x} \in R^{\mathbf{A}}$ for some relation symbol R, and we aim to show that $g(\mathbf{x}) \in R^{\mathbf{B}}$. Since $\mathbf{x} \in R^{\mathbf{A}}$, either $\mathbf{x} \in R^{\mathbf{B}}$, or $\mathbf{x} \in u(R^{\mathbf{C}})$ for some $u \in F$. In the first case, $g(\mathbf{x}) = \mathbf{x}$ and hence $g(\mathbf{x}) \in R^{\mathbf{B}}$ as required. In the second case, $\mathbf{x} = u(\mathbf{y})$ for some $\mathbf{y} \in R^{\mathbf{C}}$. Let $\mathbf{y} = (y_1, \ldots, y_r)$ and let us analyze the components y_i distinguishing by cases whether they belong to some $P_b^{\mathbf{C}}$ or not. Suppose first $y_i \in P_b^{\mathbf{C}}$ for some b. Then $h(y_i) = b$ by the remark above. Also $u(y_i) \in P_b^{\mathbf{C}} \cup \{b\}$ by the definition of F. Continuing, if $u(y_i) \in P_b^{\mathbf{C}}$ then $g(u(y_i)) = b$ again by the remark above, and if $u(y_i) = b$ then $g(u(y_i)) = g(b) = b$ by the definition of g. Therefore $g(u(y_i)) = h(y_i)$. Suppose next that $y_i \notin P_b^{\mathbf{C}}$ for all $b \in B$. Then $u(y_i) = y_i$ by the definition of F, and $g(u(y_i)) = h(y_i)$ again. It follows that $g(u(\mathbf{y})) = h(\mathbf{y})$. Since $\mathbf{y} \in R^{\mathbf{C}}$ and h is a homomorphism from **C** to **D**, we have $h(\mathbf{y}) \in R^{\mathbf{D}}$. It follows that $g(\mathbf{x}) \in R^{\mathbf{B}}$ because $g(\mathbf{x}) = g(u(\mathbf{y})) = h(\mathbf{y})$ and $R^{\mathbf{D}} = R^{\mathbf{B}}$. This proves that g is a homomorphism.

Suppose next that f is a homomorphism from **A** to **B**. Note that **B** is an induced substructure of **A**, so the restriction of f to B is a homomorphism from **B** to itself. Since **B** is a core, this restriction must be an automorphism π of **B**. We may assume then that f is the identity on B; otherwise we start with the homomorphism obtained from f by composing it with π^{-1} on B. Now we define the map $h : C \to B$ as follows: if $y \in P_b^{\mathbf{C}}$ for some $b \in B$, then $h(y) = b$; otherwise, $h(y) = f(y)$. Since we are assuming that the sets $P_b^{\mathbf{C}}$ are pairwise disjoint, the map h is well-defined. We claim that h is a homomorphism from **C** to **D**. First note that if $y \in P_b^{\mathbf{C}}$, then $h(y) \in P_b^{\mathbf{D}}$ by definition. Now, let $\mathbf{y} \in R^{\mathbf{C}}$ for some relation symbol R, and we prove $h(\mathbf{y}) \in R^{\mathbf{D}}$. Define $u : C \to A$ by $u(y) = b$ if $y \in P_b^{\mathbf{C}}$ for some b, and $u(y) = y$ otherwise. Since the sets $P_b^{\mathbf{C}}$ are disjoint, this is well-defined. Note that $u \in F$. Let $\mathbf{y} = (y_1, \ldots, y_r)$ and let us analyze the components y_i distinguishing by cases on whether they belong to some $P_b^{\mathbf{C}}$ or not. Suppose first that $y_i \in P_b^{\mathbf{C}}$ for some b. Then $u(y_i) = b$ by the definition of u, and $f(u(y_i)) = b$ because f is the identity on B. Also $h(y_i) = b$ by the definition of h. Therefore $h(y_i) = f(u(y_i))$. Suppose next that $y_i \notin P_b^{\mathbf{C}}$ for any b. Then $u(y_i) = y_i$ by the definition of u, and $h(y_i) = f(y_i)$ by the definition of h. Again $h(y_i) = f(u(y_i))$. It follows then that $h(\mathbf{y}) = f(u(\mathbf{y}))$. Now, $u(\mathbf{y}) \in R^{\mathbf{A}}$ because $u \in F$ and $\mathbf{y} \in R^{\mathbf{C}}$. Hence $f(u(\mathbf{y})) \in R^{\mathbf{B}}$ because f is a homomorphism from **A** to **B**. Thus $h(\mathbf{y}) \in R^{\mathbf{D}}$ because $R^{\mathbf{D}} = R^{\mathbf{B}}$. This proves that h is a homomorphism. □

We need to show that this reduction is indeed existential positive. Fix a pair of distinct variables p_0, p_1 that will play the role of parameters. For concreteness, we can think of p_0 and p_1 as distinct elements of C. Let $q = |B|$ and $t = \lfloor \log_2 q \rfloor + 1$. We can think of the universe of **A** as the subset of C^{t+1} defined by the following formula with $t + 1$ free variables $y_0, y_1, \ldots, y_t$:

$$(y_0 = p_0 \wedge y_1 = \cdots = y_t) \vee (y_1 = p_1 \wedge \psi(y_1, \ldots, y_t)),$$

where $\psi(y_1, \ldots, y_t)$ is a formula that is satisfied by the set of numbers $k \in \{0, \ldots, q-1\}$ when encoded in binary; the bits are encoded by $y_b = p_0$ or $y_b = p_1$. This is the same technique as in the proofs of Lemmas 12 and 17. Intuitively, the set of tuples $(y_0, \ldots, y_t)$ for which $y_0 = p_0 \wedge y_1 = \cdots = y_t$ holds encodes C, and the set of tuples for which $y_0 = p_1 \wedge \psi(y_1, \ldots, y_t)$ encodes B. Now we define the interpretation of the relation symbol R by the following formula:

$$\xi \vee \phi_R \vee \bigvee_{v \in G} \theta_{v,R},$$

where G is the set of mappings $v : \{1, \ldots, r\} \to B \times \{0, 1\}$, and ξ, ϕ_R and $\theta_{v,R}$ are formulas to be described soon. Note the similarity of this formula with the abstract definition of $R^{\mathbf{A}}$ that we gave:

$$R^{\mathbf{B}} \cup \bigcup_{u \in F} u(R^{\mathbf{C}}).$$

The formula $\phi_R(\mathbf{y}^1, \dots, \mathbf{y}^r, \mathbf{p})$ encodes the set $R^{\mathbf{B}}$ as a finite disjunction of conjunctions of equalities encoding the tuples of $R^{\mathbf{B}}$. This is easy to work out. The formula $\theta_{v,R}(\mathbf{y}^1, \dots, \mathbf{y}^r, \mathbf{p})$ encodes the set $u(R^{\mathbf{C}})$ as follows:

$$(\exists z_1) \cdots (\exists z_r)(R(z_1, \dots, z_r) \wedge T_1 \wedge T_2)$$

where

$$T_1 = \bigwedge_{j \in S_1} y_0^j = p_0 \wedge y_1^j = \cdots = y_t^j \wedge P_b(z_j) \wedge P_b(y_1^j)$$

$$T_2 = \bigwedge_{j \in S_2} y_0^j = p_1 \wedge y_1^j = p_{b_1} \wedge \cdots \wedge y_t^j = p_{b_t}$$

where $b_1, \dots, b_t$ denote the bits of the binary encoding of b in a fixed numbering of B, and where S_1 is the set of j such that $v(j) = (b, 0)$ and S_2 is the set of j such that $v(j) = (b, 1)$. Finally, the formula ξ is defined as

$$\bigvee_{b_1 \neq b_2} (\exists z)(P_{b_1}(z) \wedge P_{b_2}(z)),$$

where b_1 and b_2 range over B. This completes the definition of $R^{\mathbf{A}}$. Note that ξ is used to make $R^{\mathbf{A}} = A^r$ whenever the sets $P_b^{\mathbf{C}}$ are not disjoint. □

6. Omitting types and definability

Let $\mathcal{A}$ be an algebra. A *congruence* of $\mathcal{A}$ is an equivalence relation α that is invariant with respect to all operations of $\mathcal{A}$. In other words, for any (n-ary) operation f of $\mathcal{A}$ and any $a_1, \dots, a_n, b_1, \dots, b_n \in \mathcal{A}$ such that $(a_i, b_i) \in \alpha$ we have $(f(a_1, \dots, a_n), f(b_1, \dots, b_n)) \in \alpha$. The congruences of $\mathcal{A}$ form its *congruence lattice* $\mathrm{con}(\mathcal{A})$. A *prime quotient* in this lattice is a pair of congruences α, β such that $\alpha \leq \beta, \alpha \neq \beta$, and for any γ with $\alpha \leq \gamma \leq \beta$ we have either $\alpha = \gamma$, or $\beta = \gamma$. The fact that α, β is a prime quotient will be denoted by $\alpha \prec \beta$. For any $a \in \mathcal{A}$ we denote by a^α the equivalence class containing a. The set of all equivalence classes is called the *quotient set*, $A/_\alpha$. The fact that α is a congruence allows one to define the action of any operation of $\mathcal{A}$ on the quotient set:

$$f^\alpha(a_1^\alpha, \dots, a_n^\alpha) = (f(a_1, \dots, a_n))^\alpha.$$

The quotient set endowed with all the operations f^α is called the *quotient algebra* $\mathcal{A}/_\alpha$ of $\mathcal{A}$. It is not hard to see that $\mathcal{A}/_\alpha$ is the homomorphic image of $\mathcal{A}$ under the homomorphism that takes $a \in \mathcal{A}$ to a^α.

6.1. Unary and affine types and modules

Tame congruence theory [18] allows one to assign to each prime quotient of the congruence lattice $\mathrm{con}(\mathcal{A})$ of a finite algebra $\mathcal{A}$ one of five types. The type reflects the local structure of the algebra, which can be one of the following:

1. a finite set with a group action on it,
2. a finite vector space over a finite field,
3. a two-element Boolean algebra,
4. a two-element lattice,
5. a two-element semilattice.

In what follows we also refer to type **1** as the *unary type* and type **2** as the *affine type*. We use tame congruence as a black box extracting properties we need from existing results, and we do not therefore need a precise definition of the types. The type of a prime quotient $\alpha \prec \beta$ is denoted by $\mathsf{typ}(\alpha, \beta)$, while $\mathsf{typ}(\mathcal{A})$ denotes the set of types appearing as types of some prime quotient of $\mathcal{A}$. If $\mathfrak{A}$ is a class of algebras, $\mathsf{typ}(\mathfrak{A})$ denotes the set $\bigcup_{\mathcal{A} \in \mathfrak{A}} \mathsf{typ}(\mathcal{A})$. If $\mathbf{i} \notin \mathsf{typ}(\mathfrak{A})$, we say that $\mathfrak{A}$ *omits* type $\mathbf{i}$. Otherwise, we say $\mathfrak{A}$ *admits* type $\mathbf{i}$.

Lemma 20. *Let $\mathcal{A}$ be a finite idempotent algebra. If* $\mathrm{var}(\mathcal{A})$ *admits types* **1** *or* **2** *then it contains a finite idempotent reduct of a module.*

Proof. By a result from [6], if $\mathrm{var}(\mathcal{A})$ does not omit type **1** then it contains a finite *set*, that is an algebra all of whose operations are projections. So, suppose that $\mathrm{var}(\mathcal{A})$ omits type **1**, but does not omit type **2**.

Since $\mathrm{var}(\mathcal{A})$ does not omit type 2, there is a finite algebra $\mathcal{B} \in \mathrm{var}(\mathcal{A})$ and a prime quotient $\alpha \prec \beta \in \mathrm{con}(\mathcal{B})$ such that $\mathsf{typ}(\alpha, \beta) = \mathbf{2}$. Note first that taking $\mathcal{B}/_\alpha$ instead of $\mathcal{B}$ we may assume that $\alpha = \underline{0}$, the equality relation, because it follows from tame congruence theory that $\mathsf{typ}(\alpha/_\gamma, \beta/_\gamma) = \mathsf{typ}(\alpha, \beta)$ for any $\gamma \leq \alpha$. Next we note that since $\mathcal{B}$ is an idempotent algebra, every congruence class of β is a subalgebra. Take a non-trivial β-class, and let $\mathcal{C}$ be the corresponding subalgebra. The restriction of β to C is the total congruence $\underline{1}$.

A congruence θ *centralizes* η *modulo* ϵ if for any term operation $f(x_1, \dots, x_n, y_1, \dots, y_n, z_1, \dots, z_k)$, any $c_1, \dots, c_k \in A$, and any $a_1^1, \dots, a_n^1, a_1^2, \dots, a_n^2, b_1^1, \dots, b_m^1, b_1^2, \dots, b_m^2$ in A such that $(a_i^1, a_i^2) \in \theta$, $(b_i^1, b_i^2) \in \eta$, the following implication holds:

$$\begin{gathered} f(a_1^1, \dots, a_n^1, b_1^1, \dots, b_m^1, c_1, \dots, c_k) \overset{\epsilon}{\equiv} f(a_1^2, \dots, a_n^2, b_1^1, \dots, b_m^1, c_1, \dots, c_k) \\ \Downarrow \\ f(a_1^1, \dots, a_n^1, b_1^2, \dots, b_m^2, c_1, \dots, c_k) \overset{\epsilon}{\equiv} f(a_1^2, \dots, a_n^2, b_1^2, \dots, b_m^2, c_1, \dots, c_k). \end{gathered}$$

It is known that $\mathsf{typ}(\eta, \theta) \in \{\mathbf{1}, \mathbf{2}\}$ if and only if θ centralizes itself modulo η (see [18, Theorem 7.2]).

In our situation we have that β centralizes itself modulo $\underline{0}$ in $\mathcal{B}$. Therefore, $\underline{1}$ centralizes itself modulo $\underline{0}$ in $\mathcal{C}$. This implies $\mathsf{typ}(\mathcal{C}) \subseteq \{\mathbf{1}, \mathbf{2}\}$, and, since $\mathrm{var}(\mathcal{A})$ omits type $\mathbf{1}$, we obtain $\mathsf{typ}(\mathcal{C}) = \{\mathbf{2}\}$. By Theorem 9.6 of [18] there is a ternary term operation d that is Mal'tsev on $\mathcal{C}$, that is d satisfies the identities $d(x, y, y) = d(y, y, x) = x$. Therefore $\mathcal{C}$ generates a congruence permutable variety, and by a result of [17] it is an idempotent reduct of a module. □

Recall from Section 3 the definition of the structure $\mathbf{E}_{\mathcal{G},r}$ for every finite Abelian group $\mathcal{G}$ and every integer $r \geq 1$.

Lemma 21. *Let $\mathcal{M}$ be a finite module, let $\mathcal{G}$ be the Abelian group underlying the ring of $\mathcal{M}$, and let $\mathcal{A}$ be an idempotent reduct of $\mathcal{M}$. Then $\mathcal{A}$ is a reduct of the algebra of $\mathbf{E}_{\mathcal{G},r}$ for every $r \geq 1$.*

Proof. Let $\mathbf{E} = \mathbf{E}_{\mathcal{G},r}$. Every m-ary term operation of $\mathcal{A}$ can be represented in the form

$$f(x_1, \dots, x_m) = r_1 x_1 + \cdots + r_m x_m,$$

and, as f is idempotent, $r_1 + \cdots + r_m = 1$. Take m tuples $\mathbf{a}_1, \dots, \mathbf{a}_m$ in the relation R_a^j in $\mathbf{E}$, where $\mathbf{a}_i = (a_{i1}, \dots, a_{ij})$ for $i \in \{1, \dots, m\}$. Check that the tuple

$$(f(a_{11}, \dots, a_{m1}), \dots, f(a_{1j}, \dots, a_{mj}))$$

also belongs to R_a^j:

$$\begin{aligned} & f(a_{11}, \dots, a_{m1}) + \cdots + f(a_{1j}, \dots, a_{mj}) \\ & \quad = (r_1 a_{11} + \cdots + r_j a_{m1}) + \cdots + (r_1 a_{1j} + \cdots + r_m a_{mj}) \\ & \quad = r_1(a_{11} + \cdots + a_{1j}) + \cdots + r_m(a_{m1} + \cdots + a_{mj}) \\ & \quad = r_1 a + \cdots + r_m a \\ & \quad = a. \end{aligned}$$

Therefore, every relation of $\mathbf{E}$ is invariant under every operation of $\mathcal{A}$. That is, $\mathcal{A}$ is a reduct of the algebra of $\mathbf{E}$. □

6.2. Unary and affine types and definability

We can bring together the results of Section 5 and the previous subsection to establish the following theorem.

Theorem 22. *Let $\mathbf{B}$ be a finite structure and let $\mathcal{B}$ be its algebra. If $\mathrm{var}(\mathcal{B})$ admits types $\mathbf{1}$ or $\mathbf{2}$ then there exists a non-trivial finite Abelian group $\mathcal{G}$ such that $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},r}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{B})$ for every $r \geq 1$.*

Proof. Since $\mathrm{CSP}(\mathbf{B}) = \mathrm{CSP}(\mathrm{core}(\mathbf{B}))$, where $\mathrm{core}(\mathbf{B})$ is the core of $\mathbf{B}$, we may assume that $\mathbf{B}$ is a core. Let $\mathbf{D}$ be the singleton-expansion of $\mathbf{B}$ and let $\mathcal{D}$ be its algebra, which is idempotent. By Lemma 19, we have $\mathrm{CSP}(\mathbf{D}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{B})$. Moreover, if $\mathrm{var}(\mathcal{B})$ admits types $\mathbf{1}$ or $\mathbf{2}$, so does $\mathrm{var}(\mathcal{D})$ because $\mathcal{D}$ is a reduct of $\mathcal{B}$ (see [18, Chapter 5]). By Lemma 20, the variety $\mathrm{var}(\mathcal{D})$ contains a finite idempotent reduct $\mathcal{A}$ of a module. Let $\mathcal{G}$ be the Abelian group underlying the ring of the module. Then $\mathcal{G}$ is non-trivial and finite. Moreover, $\mathcal{A}$ is a reduct of the algebra of $\mathbf{E}_{\mathcal{G},r}$ for every $r \geq 1$ by Lemma 21. It follows that $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},r}) \leq_{\mathrm{datalog}} \mathrm{CSP}(\mathbf{D})$. Composing we get the result. □

We have seen in Section 3 that $\mathrm{CSP}(\mathbf{E}_{\mathcal{G},3})$ is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$ when $\mathcal{G}$ is non-trivial. Since definability in $\mathrm{C}^{\omega}_{\infty\omega}$ is preserved downwards by Datalog-reductions, this yields the following corollary:

Corollary 23. *Let $\mathbf{B}$ be a finite structure and let $\mathcal{B}$ be its algebra. If $\mathrm{CSP}(\mathbf{B})$ is definable in $\mathrm{C}^{\omega}_{\infty\omega}$, then $\mathrm{var}(\mathcal{B})$ omits types $\mathbf{1}$ and $\mathbf{2}$.*

Corollary 23 can be seen as a strengthening of the result of Larose and Zadori [24] that if the complement of $\mathrm{CSP}(\mathbf{B})$ is definable in Datalog then $\mathrm{var}(\mathcal{B})$ omits types $\mathbf{1}$ and $\mathbf{2}$. Larose and Zadori also conjectured the converse, namely that if $\mathrm{var}(\mathcal{B})$ omits types $\mathbf{1}$ and $\mathbf{2}$ then the complement of $\mathrm{CSP}(\mathbf{B})$ is definable in Datalog. By Corollary 23 this conjecture would imply that every $\mathrm{CSP}(\mathbf{B})$ is either definable in Datalog or not definable in $\mathrm{C}^{\omega}_{\infty\omega}$, which can be seen as a definability dichotomy.

Another consequence of Corollary 23 is that graph 3-colourability (i.e., $\mathrm{CSP}(\mathbf{K}_3)$) is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$, since its algebra has no operations but the projections and therefore admits type $\mathbf{1}$.

Corollary 24. *Graph 3-colourability is not definable in $\mathrm{C}^{\omega}_{\infty\omega}$.*

While this has previously been proved directly [11], our result gives a new proof that gives an algebraic explanation for why the problem is not definable.

References

[1] A. Blass, Y. Gurevich, S. Shelah, On polynomial time computation over unordered structures, Journal of Symbolic Logic 67 (3) (2002) 1093–1125.

[2] V.G. Bodnarchuk, L.A. Kaluzhnin, V.N. Kotov, B.A. Romov, Galois theory for post algebras. i, Kibernetika 3 (1969) 1–10 (in Russian).

[3] A.A. Bulatov, V. Dalmau, A simple algorithm for Mal'tsev constraints, SIAM Journal on Computing 36 (1) (2006) 16–27.

[4] A.A. Bulatov, P. Jeavons, A. Krokhin, Classifying the complexity of constraints using finite algebras, SIAM Journal on Computing 34 (2005) 720–742.

[5] A.A. Bulatov, Mal'tsev constraints are tractable, Technical Report PRG-RR-02-05, Computing Laboratory, University of Oxford, Oxford, UK, 2002.

[6] A.A. Bulatov, P.G. Jeavons, Algebraic structures in combinatorial problems, Technical Report MATH-AL-4-2001, Technische universität Dresden, Dresden, Germany, 2001.

[7] G. Buntrock, C. Damm, U. Hertrampf, C. Meinel, Structure and importance of logspace-MOD class, Mathematical Systems Theory 25 (1992) 223–237.

[8] S. Burris, H.P. Sankappanavar, A Course in Universal Algebra, in: Graduate Texts in Mathematics, vol. 78, Springer-Verlag, New York, Berlin, 1981.

[9] V. Dalmau, Linear datalog and bounded path duality of relational structures, Logical Methods in Computer Science 1 (1) (2005).

[10] V. Dalmau, Ph.G. Kolaitis, M.Y. Vardi, Constraint satisfaction, bounded treewidth, and finite variable logics, in: Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming, CP'02, in: Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 311–326.

[11] A. Dawar, A restricted second order logic for finite structures, Information and Computation 143 (1998) 154–174.

[12] R. Diestel, Graph Theory, Springer, 1997.

[13] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, 2nd ed., Springer, 1999.

[14] T. Feder, M.Y. Vardi, Computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory, SIAM Journal of Computing 28 (1998) 57–104.

[15] D. Geiger, Closed systems of function and predicates, Pacific Journal of Mathematics (1968) 95–100.

[16] L. Hella, Logical hierarchies in PTIME, Information and Computation 129 (1996) 1–19.

[17] C. Herrman, Affine algebras in congruence-modular varieties, Acta Scientiarum Mathematicarum (Szeged) 41 (1971) 119–125.

[18] D. Hobby, R.N. Mckenzie, The Structure of Finite Algebras, in: Contemporary Mathematics, vol. 76, American Mathematical Society, Providence, R.I, 1988.

[19] N. Immerman, Descriptive Complexity, Springer, 1999.

[20] P.G. Jeavons, D.A. Cohen, M. Gyssens, Closure properties of constraints, Journal of the ACM 44 (1997) 527–548.

[21] P.G. Jeavons, D.A. Cohen, J.K. Pearson, Constraints and universal algebra, Annals of Mathematics and Artificial Intelligence 24 (1998) 51–67.

[22] Ph.G. Kolaitis, M.Y. Vardi, A game-theoretic approach to constraint satisfaction, in: Proc. 17th National Conference on Artificial Intelligence, AAAI-2000, 2000, pp. 175–181.

[23] B. Larose, C. Loten, C. Tardif, A characterisation of first-order constraint satisfaction problems, in: Proc. 21st IEEE Symp. on Logic in Computer Science, 2006, pp. 201–210.

[24] B. Larose, L. Zádori, Bounded width problems and algebras, Algebra Universalis 58 (2007) 439–466.

[25] L. Libkin, Elements of Finite Model Theory, Springer, 2004.

[26] A.K. Mackworth, Consistency in networks of relations, Artificial Intelligence 8 (1977) 99–118.

[27] U. Montanari, Networks of constraints: Fundamental properties and applications to picture processing, Information Sciences 7 (1974) 95–132.

[28] O. Reingold, Undirected st-connectivity in log-space, in: 37th Annual Symposium on Theory of Computing, 2005, pp. 376–385.

[29] P. Seymour, R. Thomas, Graph searching and a min–max theorem for treewidth, Journal of Combinatorial Theory, Series B 58 (1993) 22–33.