



## Universal algebra and hardness results for constraint satisfaction problems<sup>☆</sup>

Benoît Larose<sup>a</sup>, Pascal Tesson<sup>b,\*</sup>

<sup>a</sup> Department of Mathematics and Statistics, Concordia University, Canada

<sup>b</sup> Département d'Informatique et de Génie Logiciel, Université Laval, Canada

### ARTICLE INFO

#### Keywords:

Constraint satisfaction  
Universal algebra  
Complexity classes  
Datalog

### ABSTRACT

We present algebraic conditions on constraint languages  $\Gamma$  that ensure the hardness of the constraint satisfaction problem  $\text{CSP}(\Gamma)$  for complexity classes L, NL, P, NP and  $\text{Mod}_p\text{L}$ . These criteria also give non-expressibility results for various restrictions of Datalog. Furthermore, we show that if  $\text{CSP}(\Gamma)$  is not first-order definable then it is L-hard. Our proofs rely on tame congruence theory and on a fine-grain analysis of the complexity of reductions used in the algebraic study of CSP. The results pave the way for a refinement of the dichotomy conjecture stating that each  $\text{CSP}(\Gamma)$  lies in P or is NP-complete and they match the recent classification of [E. Allender, M. Bauland, N. Immerman, H. Schnoor, H. Vollmer, The complexity of satisfiability problems: Refining Schaefer's theorem, in: Proc. 30 th Math. Found. of Comp. Sci., MFCS'05, 2005, pp. 71–82] for Boolean CSP. We also infer a partial classification theorem for the complexity of  $\text{CSP}(\Gamma)$  when the associated algebra of  $\Gamma$  is the full idempotent reduct of a preprimal algebra.

© 2009 Published by Elsevier B.V.

Constraint satisfaction problems (CSP) provide a unifying framework to study various computational problems arising naturally in artificial intelligence, combinatorial optimization, graph homomorphisms and database theory. An instance of this problem consists of a finite domain, a list of variables and constraints relating the possible values of variables: one has to decide whether the variables can be assigned values that simultaneously satisfy all constraints. This problem is of course NP-complete and so research has focused on identifying tractable subclasses of CSP. A lot of attention has been given to the case where all constraints are constructed from some *constraint language*  $\Gamma$ , i.e. some set of finitary relations over a fixed domain. In an instance of  $\text{CSP}(\Gamma)$ , all constraints are of the form  $(x_{i_1}, \dots, x_{i_k}) \in R_j$  for some  $R_j \in \Gamma$ .

In their seminal work [15], Feder and Vardi conjectured that each  $\text{CSP}(\Gamma)$  either lies in P or is NP-complete. This so-called *dichotomy conjecture* is the natural extension to non-Boolean domains of a celebrated result of Schaefer [28] on the complexity of Generalized Satisfiability which states that  $\text{CSP}(\Gamma)$  is either in P or is NP-complete for any constraint language  $\Gamma$  over the Boolean domain.

Progress towards the dichotomy conjecture has been steady over the last fifteen years and has been driven by a number of complementary approaches. One angle of attack relies on universal algebra: there is a natural way to associate to a set of relations  $\Gamma$  an algebra  $\mathbb{A}(\Gamma)$  whose operations are the functions that *preserve* the relations in  $\Gamma$  and one can show that the complexity of  $\text{CSP}(\Gamma)$  depends on the algebraic structure of  $\mathbb{A}(\Gamma)$ . This analysis has led to a number of key results including a verification of the dichotomy conjecture for three-element domains [5] and for so-called list-CSP [6], as well as the identification of wide classes of tractable CSP (see [9]).

<sup>☆</sup> Research supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds Québécois pour la Recherche sur la Nature et les Technologies (FQRNT) and Montreal's Centre de Recherches Mathématiques (CRM).

\* Corresponding author.

E-mail addresses: [larose@mathstat.concordia.ca](mailto:larose@mathstat.concordia.ca) (B. Larose), [pascal.tesson@ift.ulaval.ca](mailto:pascal.tesson@ift.ulaval.ca) (P. Tesson).

A different, descriptive complexity approach has consisted in classifying CSP according to the sophistication of the logical apparatus required to define the set of negative instances of  $\text{CSP}(\Gamma)$ . It was noted early on that when  $\neg\text{CSP}(\Gamma)$  is definable in the database-inspired logic programming language Datalog then  $\text{CSP}(\Gamma)$  lies in P [15] and this provides a unifying explanation for a number of (but not all) tractable cases. Further investigations have indicated strong connections between expressibility in symmetric Datalog and linear Datalog, and CSP solvability in, respectively, logarithmic space (L) and non-deterministic logspace (NL) [11,13]. From a logical perspective, the simplest class of CSP are those which are first-order definable and recent work has provided a precise characterization for them [3,22].

From a complexity-theoretic perspective, the classification of  $\text{CSP}(\Gamma)$  as “tractable” is rather coarse. Ultimately, one would expect that every  $\text{CSP}(\Gamma)$  lying in P is in fact complete for some “fairly standard” subclass of P. Over the two-element domain, it was recently established that Schaefer’s dichotomy can indeed be refined: each  $\text{CSP}(\Gamma)$  over the Boolean domain is either **FO**-definable or is complete under  $\text{AC}^0$  isomorphisms for one of the classes L, NL,  $\oplus\text{L}$ , P or NP [2].

The present paper seeks to develop the necessary tools for a refinement of the dichotomy conjecture and for a smoother integration of the logical and algebraic approaches to the study of CSP. As we recall in Section 2, the algebraic angle of attack relies on a number of basic reductions from  $\text{CSP}(\Lambda)$  to  $\text{CSP}(\Gamma)$  when  $\Gamma$  and  $\Lambda$  are constraint languages such that the algebra  $\mathbb{A}(\Lambda)$  lies in the variety generated by  $\mathbb{A}(\Gamma)$  [7,9]. When the sole objective is to classify  $\text{CSP}(\Gamma)$  as either in P or NP-complete, polynomial-time Turing reductions are clearly good-enough. However, finer classifications require much tighter reductions and we show that all but one of these basic reductions is in fact first-order. Furthermore, we show that all of them preserve expressibility in Datalog and its most relevant fragments.

These reductions provide the opportunity for a systematic study of the complexity of tractable CSP. In Section 3, we begin by proving that if  $\text{CSP}(\Gamma)$  is not first-order definable, then the problem is in fact L-hard under first-order reductions. This result can be viewed as a first step towards more general dichotomy theorems as it exhibits a gap in the complexity of CSP. In fact, our argument shows that  $\text{CSP}(\Gamma)$  is L-hard if  $\Gamma$  does not have finite duality and, as a corollary, we can use circuit-complexity lower bounds to re-prove Atserias’ characterization of first-order definable CSP [3] and strengthen its scope.

In Section 4, we use tame congruence theory and deep classification results of idempotent strictly simple algebras to obtain a number of hardness results for  $\text{CSP}(\Gamma)$ . Specifically, we consider the variety  $\mathcal{V}$  generated by  $\mathbb{A}(\Gamma)$  and give sufficient conditions on  $\mathcal{V}$  for  $\text{CSP}(\Gamma)$  to be NL-hard,  $\text{Mod}_p\text{L}$ -hard and P-hard. These also translate into necessary conditions for  $\neg\text{CSP}(\Gamma)$  to be definable in Datalog, linear Datalog and symmetric Datalog. For a given  $\Gamma$  it is possible to decide whether or not **V** fits each of these criteria [31]. In Section 5, we demonstrate the usefulness of the results by revisiting the results of [2] on CSP over the boolean domain and by classifying the complexity of a number of  $\text{CSP}(\Gamma)$  when  $\mathbb{A}(\Gamma)$  is the full idempotent reduct of a preprimal algebra.

An extended abstract of the present paper appeared in the proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP’07).

## 1. Preliminaries

We first set the notation and present the required basics. We refer the reader to [18] and [12] for algebraic and clone-theoretic results, to [19] for an introduction to finite model theory and descriptive complexity and to [9] for a survey on the algebraic approach to CSP.

Let  $\sigma = \{R_1, \dots, R_r, c_1, \dots, c_s\}$  be a signature, where each  $R_i$  is a relational symbol of arity  $a_i$  and each  $c_i$  is a constant symbol. A structure  $\mathbf{H}$  of signature  $\sigma$  is a tuple  $\mathbf{H} = \langle H; R_1(\mathbf{H}), \dots, R_r(\mathbf{H}), c_1(\mathbf{H}), \dots, c_s(\mathbf{H}) \rangle$  where  $H$ , the *universe* of  $\mathbf{H}$ , is a non-empty set, and for each  $i$ ,  $R_i(\mathbf{H})$  is a relation on  $H$  of arity  $a_i$ , and  $c_i(\mathbf{H})$  is some fixed element of  $H$ . We follow the standard convention of using  $G, H, \dots$  to denote the universe of the structure  $\mathbf{G}, \mathbf{H}, \dots$ . Unless otherwise mentioned the signatures we deal with in this paper are purely relational (i.e. without constant symbols).

Let  $\sigma$  be a (relational) signature. Given two  $\sigma$ -structures  $\mathbf{G}$  and  $\mathbf{H}$ , a map  $f$  from  $G$  to  $H$  is a *homomorphism from  $\mathbf{G}$  to  $\mathbf{H}$*  if  $f(R_i(\mathbf{G})) \subseteq R_i(\mathbf{H})$  for all  $i$ , where for any relation  $R$  of arity  $r$  we have

$$f(R) = \{(f(x_1), \dots, f(x_r)) : (x_1, \dots, x_r) \in R\}.$$

Two structures  $\mathbf{H}$  and  $\mathbf{H}'$  are *homomorphically equivalent* if there exist homomorphisms  $\mathbf{H} \rightarrow \mathbf{H}'$  and  $\mathbf{H}' \rightarrow \mathbf{H}$ . A structure  $\mathbf{H}$  is a *core* if the only homomorphisms  $\mathbf{H} \rightarrow \mathbf{H}$  are automorphisms, or, equivalently, if it is of minimal size in its class of homomorphically equivalent structures. Every finite structure is equivalent to a structure of minimal size, and it is easy to verify that any two minimal structures are isomorphic, hence we can talk about *the core* of a structure.

Let  $\Gamma$  be a *constraint language*, i.e. a finite set of relations on the set  $H$ . Let  $\mathbf{\Gamma}$  denote a relational structure on  $H$  whose set of basic relations is  $\Gamma$  and let  $\sigma$  be its signature. We denote as  $\text{Hom}(\mathbf{\Gamma})$  the class of all finite structures of type  $\sigma$  that admit a homomorphism to  $\mathbf{\Gamma}$ . In this setting  $\mathbf{\Gamma}$  is called the *target structure*. Alternatively, we may use the notation  $\text{CSP}(\Gamma)$  for this decision problem: indeed the constraints in an instance of  $\text{CSP}(\Gamma)$  can be regarded as defining a  $\sigma$ -structure  $\mathbf{C}$  on the set of variables and a satisfying assignment is a homomorphism from  $\mathbf{C}$  to  $\mathbf{\Gamma}$ . We thus occasionally abuse notation and write  $\text{CSP}(\mathbf{\Gamma})$  to denote  $\text{Hom}(\mathbf{\Gamma})$ . Obviously, if  $\mathbf{\Gamma}'$  is the core of  $\mathbf{\Gamma}$  then  $\text{CSP}(\mathbf{\Gamma}) = \text{CSP}(\mathbf{\Gamma}')$ . We thus assume throughout the paper that the target structures under consideration are cores.

### 1.1. Algebras and varieties

An  $n$ -ary operation on a set  $A$  is a map from  $A^n$  to  $A$ . The  $n$ -ary operation  $f$  on  $A$  preserves the  $k$ -ary relation  $\theta$  on  $A$  (equivalently, we say that  $\theta$  is *invariant under  $f$* ) if the following holds: given any matrix  $M$  of size  $k \times n$  whose columns are in  $\theta$ , applying  $f$  to the rows of  $M$  will produce a  $k$ -tuple in  $\theta$ . Given a set  $\Gamma$  of relations on  $A$ ,  $\text{Pol}(\Gamma)$  denotes<sup>1</sup> the set of all operations on  $A$  that preserve all relations in  $\Gamma$ .

An *algebra* is a pair  $\mathbb{A} = \langle A; F \rangle$  where  $A$  is a non-empty set, called the universe of  $\mathbb{A}$ , and  $F$  is a set of operations on  $A$ , called the basic operations of  $\mathbb{A}$ . For a constraint language  $\Gamma$  over  $A$ , we denote by  $\mathbb{A}(\Gamma)$  the algebra  $(A; \text{Pol}(\Gamma))$ , and call it the *algebra associated to CSP* ( $\Gamma$ ). The algebraic approach to CSP [9] relies on the very strong link between the complexity of  $\text{CSP}(\Gamma)$  and the algebraic properties of  $\mathbb{A}(\Gamma)$  (see Theorem 2.1).

The *term operations* of an algebra are the operations that can be built from its basic operations using composition and projections. Similarly, the *polynomial operations* are those built from the basic operations using constants, composition and projections. Two algebras are *term (polynomially) equivalent* if they have the same term (polynomial) operations. An operation  $f$  is *idempotent* if it satisfies  $f(x, \dots, x) = x$  for all  $x$ . The *(full) idempotent reduct* of the algebra  $\mathbb{A}$  is the algebra with the same universe and whose basic operations are the idempotent terms of  $\mathbb{A}$ .

Subalgebras, homomorphic images and products of algebras are defined in a natural way, as for groups or rings. Technically we require the algebras to be indexed and of the same signature to define these notions (see [18]). A class of similar algebras (i.e. algebras with the same signature) is a *variety* if it is closed under formation of homomorphic images (H), subalgebras (S) and products (P). The *variety generated by  $\mathbb{A}$*  is denoted by  $\mathcal{V}(\mathbb{A})$ ; it is known that  $\mathcal{V}(\mathbb{A}) = \text{HSP}(\mathbb{A})$ , i.e. that every member  $\mathbb{C}$  of the variety is obtained as a homomorphic image of a subalgebra of a power of  $\mathbb{A}$ ; furthermore this power can be taken to be finite if  $\mathbb{C}$  is finite.

Tame Congruence Theory, developed by Hobby and McKenzie [18], is a powerful tool for the analysis of finite algebras. Every finite algebra has a *typeset*, which describes the local behaviour of the algebra. It contains one or more of the following 5 types: (1) the unary type, (2) the affine type, (3) the Boolean type, (4) the lattice type and (5) the semilattice type. There is a very tight connection between the kind of equations that are satisfied by the algebras in a variety and the types that are admitted or omitted by a variety, i.e. those types that do or do not appear in the typeset of some algebra in the variety. The theory for idempotent algebras is somewhat more streamlined, and we now present the two results we shall require.

An algebra is *strictly simple* if it is simple (i.e., has no non-trivial congruences) and has no non-trivial subalgebras (a subalgebra is trivial if it is either the algebra itself or is 1-element.) A simple algebra, and in particular a strictly simple algebra, has a unique type from 1 to 5 associated to it.

Szendrei has characterised all idempotent strictly simple algebras, ([29] Theorem 6.1): we need the following special cases. The 2-element set is the 2-element algebra with no basic operations  $\langle \{0, 1\}; \emptyset \rangle$ . The 2 element semilattices are the 2-element algebras with a single binary operation  $\langle \{0, 1\}; \wedge \rangle$  and  $\langle \{0, 1\}; \vee \rangle$ . The 2 element lattice is the 2 element algebra with two binary operations  $\langle \{0, 1\}; \vee, \wedge \rangle$ . An algebra is *affine* if there is an abelian group structure on its base set such that (i)  $m(x, y, z) = x - y + z$  is a term of the algebra and (ii) every term of the algebra is affine, i.e. commutes with the operation  $m$ . Equivalently, an idempotent algebra is affine iff it is the full idempotent reduct of a module.

**Lemma 1.1** ([29]). *Let  $\mathbb{A}$  be a strictly simple idempotent algebra.*

- If  $\mathbb{A}$  has unary type (type 1) then it is term equivalent to the 2-element set;
- If  $\mathbb{A}$  has affine type (type 2) then it is an affine algebra;
- if  $\mathbb{A}$  has semilattice type (type 5) then it is term equivalent to a 2 element semilattice;
- if  $\mathbb{A}$  has lattice type (type 4) it is polynomially equivalent to the 2 element lattice.

This can be used to obtain:

**Corollary 1.2.** *Let  $\mathbb{A}$  be a finite, idempotent, strictly simple algebra.*

1. If  $\mathbb{A}$  has affine type, then there exists an Abelian group structure on the base set of  $\mathbb{A}$  such that the relation  $\{(x, y, z) : x + y = z\}$  is a subalgebra of  $\mathbb{A}^3$ ;
2. if  $\mathbb{A}$  has semilattice type, then up to isomorphism the universe of  $\mathbb{A}$  is  $\{0, 1\}$  and the relation  $\theta = \{(x, y, z) : (y \wedge z) \rightarrow x\}$  is a subalgebra of  $\mathbb{A}^3$ ;
3. if  $\mathbb{A}$  has lattice type, then up to isomorphism the universe of  $\mathbb{A}$  is  $\{0, 1\}$  and the relation  $\leq = \{(0, 0), (0, 1), (1, 1)\}$  is a subalgebra of  $\mathbb{A}^2$ .

The next lemma, due to Valeriote, will allow us to connect typesets of varieties to the complexity of CSP's:

**Lemma 1.3** (Lemma 3.1 [31]). *Let  $\mathbb{A}$  be a finite, idempotent algebra such that  $\mathcal{V}(\mathbb{A})$  admits type  $i$ . Then there exists a strictly simple algebra of type at most  $i$  in  $\text{HS}(\mathbb{A})$  where "at most  $i$ " refers to the ordering  $1 < 2 < 3 > 4 > 5 > 1$ .*

### 1.2. Fragments of datalog

Datalog was originally introduced as a database query language [30]. We view it here simply as a means to define sets of  $\sigma$ -structures. A Datalog program over the signature  $\sigma$  consists of a finite set of *rules* of the form  $h \leftarrow b_1 \wedge \dots \wedge b_k$  where each

<sup>1</sup> The notation stems from the fact that operations preserving the relations of  $\Gamma$  are sometimes known as *polymorphisms*.

of the  $b_i$  and  $h$  are atomic formulas of the form  $R(x_{j_1}, \dots, x_{j_r})$ . We distinguish two types of relational predicates occurring in the program: predicates  $R$  that occur at least once in the head (i.e. left-hand side) of a rule are called *intensional database predicates* (IDBs) and are not part of  $\sigma$ . The other predicates which occur only in the body (i.e. right-hand side) of a rule are called *extensional database predicates* (EDBs) and must all lie in  $\sigma$ . Precise definitions of the semantics of Datalog can be found in [21,11,13]: we simply illustrate the basics of the formalism with the following example.

Let  $\sigma$  be a signature consisting of a single binary relational symbol  $E$  (so that a  $\sigma$ -structure is a graph) and consider the Datalog program  $\mathcal{Q}$  consisting of the rules

$$P(x, y) \leftarrow E(x, y) \tag{1}$$

$$P(x, y) \leftarrow P(x, z) \wedge P(z, y) \tag{2}$$

$$G(x) \leftarrow P(x, x). \tag{3}$$

The program  $\mathcal{Q}$  is providing a recursive specification of the IDB predicates  $P, G$  in terms of  $E, P$  and  $G$ . The predicate  $P$  is intended to include  $(x, y)$  if there is a path from  $x$  to  $y$ . The first rule states that this holds if  $(x, y)$  is an edge and the second that, recursively, this holds if there is a path from  $x$  to some  $z$  and from  $z$  to  $y$ . The predicate  $G$  then contains the  $x$  such that there is a directed cycle around  $x$ . One of the IDBs of the Datalog program is chosen as a *target IDB* and we say that a  $\sigma$ -structure is accepted by the program if the target IDB is non-empty. The program  $\mathcal{Q}$  with  $G$  as its goal thus defines the set of graphs with a directed cycle.

Rules which contain only EDBs in their body (such as (1) above) are called *non-recursive rules* and those containing at most one IDB in their body (such as (1) and (3)) are *linear*. Although the above example contains the non-linear rule (2), it is easy to see that an equivalent linear program  $\mathcal{Q}'$  could be obtained by replacing (2) with

$$P(x, y) \leftarrow P(x, z) \wedge E(z, y).$$

A program is said to be *symmetric* if it is linear and if each recursive rule  $\mathcal{R}$  is accompanied by its *symmetric complement*  $\mathcal{R}'$ , where  $\mathcal{R}'$  is obtained from  $\mathcal{R}$  by exchanging the roles of the IDBs in its head and body. The symmetric complement of the above rule would be

$$P(x, z) \leftarrow P(x, y) \wedge E(z, y).$$

The addition of this rule to  $\mathcal{Q}'$  would yield a symmetric Datalog program which defines the set of graphs with an undirected cycle.

The expressive power of Datalog and its linear, symmetric and non-recursive fragments have been important tools in the study of CSP. A very nice result of [3] shows that  $\text{CSP}(\Gamma)$  is definable by a first-order sentence iff  $\neg\text{CSP}(\Gamma)$  is definable by a Datalog program consisting only of non-recursive rules, and consequently all such  $\text{CSP}(\Gamma)$  are solvable in co-NLOGTIME (see, e.g. [2] for a formal description of the latter class). Moreover, expressibility of  $\neg\text{CSP}(\Gamma)$  in symmetric, linear and general Datalog is a sufficient (and wide-encompassing) condition for  $\text{CSP}(\Gamma)$  to lie in respectively L [13], NL [11] and P [15].

### 1.3. Reductions and complexity

We assume basic familiarity with fundamental concepts of complexity theory, including the notions of completeness, polynomial-time and logspace reductions as well as complexity classes such as  $\text{AC}^0$ , logarithmic space (L), non-deterministic logspace (NL), polynomial-time (P) and non-deterministic polynomial-time (NP). We also refer in the sequel to the somewhat more esoteric classes  $\text{Mod}_p\text{L}$ . For a prime  $p$ ,  $\text{Mod}_p\text{L}$  denotes the class of languages recognized by  $\text{Mod}_p$ -counting non-deterministic logspace machines. Formally,  $K \in \text{Mod}_p\text{L}$  if there exists a non-deterministic logspace machine  $M$  such that  $w \in K$  iff the number of accepting paths of  $M$  on  $w$  is divisible by  $p$ . When  $p = 2$ , the corresponding class is usually denoted as  $\oplus\text{L}$ . These classes contain a number of natural problems related to modular arithmetic such as solving systems of linear equations over  $\mathbb{Z}_p$  [8]. In particular, it can be shown that solving systems of linear equations modulo  $p^k$  is also in  $\text{Mod}_p\text{L}$  and so the  $\text{CSP}(\Gamma)$  analyzed for the affine case of Theorem 5.3 in the last section is indeed  $\text{Mod}_p\text{L}$ -complete.

When proving a  $\mathcal{C}$ -hardness result, it is always desirable to establish it under the simplest possible reductions. In particular,  $\mathcal{C}$ -hardness results under polynomial-time reductions are meaningless when  $\mathcal{C}$  is a subclass of P. In this paper, we use either logspace reductions or first-order reductions (see [19] for further discussion of FO reductions). The classes L, NL,  $\text{Mod}_p\text{L}$ , P and NP are all closed under these reductions and  $\text{AC}^0$  is closed under FO-reductions.

We recall that the problems undirected  $st$ -connectivity,<sup>2</sup> directed  $st$ -connectivity,  $\text{HORN-3SAT}$  and  $\text{3SAT}$  are complete under FO-reductions for the classes L, NL, P and NP respectively [19]. There are also natural complete problems for the  $\text{Mod}_p\text{L}$ -classes. Let  $p$  be any prime number:  $\text{NOT } p\text{-GAP}$  is the set of digraphs with two distinguished nodes  $s$  and  $t$  such that the number of simple paths from  $s$  to  $t$  is divisible by  $p$ . We assume that these structures are ordered, in such a way that if  $(i, j)$  is an edge then  $i \leq j$ , and such that  $1 = s$  and  $n = t$  where  $n$  is the largest element of the universe. Note that we may assume throughout that all digraphs considered are without loops: indeed, given any digraph  $\mathbf{G}$  with edge relation  $\theta(\mathbf{G})$  we may “remove” all loops by defining a new digraph  $\mathbf{G}'$  on the same universe with edge relation  $\theta(\mathbf{G}')$  defined by the first-order formula  $\phi_\theta(x, y) \equiv (x, y) \in \theta \wedge x \neq y$ . Obviously the number of paths between  $s$  and  $t$  remains the same.

<sup>2</sup> Reingold recently proved that  $st$ -connectivity belongs to logspace [26].

**Lemma 1.4.** *The problem NOT  $p$ -GAP is mod- $p$  L complete under first-order reductions.*

This result is more or less folklore, although most completeness proofs use  $\text{NC}^1$  reductions. The reduction can be made FO using a straightforward adaptation of the NL-completeness of  $st$ -connectivity under FO reductions. (Theorem 3.16 of [19])

## 2. Nature of the algebraic and clone-theoretic reductions

The following theorem is our starting point for a fine-grained analysis of the complexity of constraint satisfaction problems. A relation  $\theta$  is *irredundant* if for each two distinct coordinates  $i$  and  $j$  there exists a tuple  $\bar{x}$  of  $\theta$  with  $x_i \neq x_j$ .

**Theorem 2.1.** *Let  $\Gamma$  be a finite set of relations on  $A$  such that  $\Gamma$  is a core. Let  $\mathbb{A}$  denote the full idempotent reduct of the algebra associated to  $\Gamma$ .*

1. *Let  $\mathbb{C}$  be a finite algebra in  $\mathcal{V}(\mathbb{A})$ , and let  $\Gamma_0$  be a finite set of relations invariant under the basic operations of  $\mathbb{C}$ . Then there exists a logspace many-one reduction of  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma)$ . Furthermore, if  $\neg\text{CSP}(\Gamma)$  is expressible in (linear, symmetric) Datalog, then so is  $\neg\text{CSP}(\Gamma_0)$ .*
2. *If furthermore  $\mathbb{C} \in \text{HS}(\mathbb{A})$  and the relations in  $\Gamma_0$  are irredundant, then the above reduction is first-order and without ordering.<sup>3</sup>*

We obtain the proof through a series of lemmas established in this section. The arguments, although not conceptually difficult, are technical and tedious. The reader can choose to initially skip Section 2 and to use the above theorem as a black box. The constructions are given for ten basic reductions: their principles are not new [20,7] although most were never explicitly shown to be first order or to preserve expressibility in the linear and symmetric fragments of Datalog. In earlier work, the reductions were shown to preserve expressibility in Datalog [24] and independent work of Atserias, Bulatov and Dawar ([4], which also appeared in the proceedings of ICALP'07) showed that they preserve important fragments of infinitary logic. It should be noted that logspace reductions are the best we can hope for in the first half of the statement: indeed, it is straightforward from the definitions to see that if  $\Gamma_0 = \Gamma \cup \{=\}$  then one has  $\mathbb{A}(\Gamma) = \mathbb{A}(\Gamma_0)$ . But if  $\text{CSP}(\Gamma)$  is first-order definable then  $\text{CSP}(\Gamma_0)$  is L-complete (see e.g. [13]) and so there can be no first-order reduction of  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma)$ .

We first introduce some terminology. We shall use the notion of *first-order interpretation with parameters* which Atserias used in a similar context in [3,4].

**Definition.** Let  $\sigma$  and  $\tau = (R_1, \dots, R_s)$  be two relational vocabularies. A  $k$ -ary first-order interpretation with  $p$  parameters of  $\tau$  in  $\sigma$  is an  $(s+1)$ -tuple  $\mathcal{I} = (\phi_U, \phi_{R_1}, \dots, \phi_{R_s})$  of first-order formulas over the vocabulary  $\sigma$ , where  $\phi_U = \phi_U(x, y)$  has  $k+p$  free variables  $x = (x^1, \dots, x^k)$  and  $y = (y^1, \dots, y^p)$  and  $\phi_{R_i} = \phi_{R_i}(x_1, \dots, x_r, y)$  has  $kr+p$  free variables where  $r$  is the arity of  $R_i$  and each  $x_j = (x_j^1, \dots, x_j^k)$  and  $y = (y^1, \dots, y^p)$ .

Let  $\mathbf{G}$  be a  $\sigma$ -structure. A tuple  $c = (c_1, \dots, c_p)$  of elements of  $\mathbf{G}$  is said to be *proper* if  $c_i \neq c_j$  when  $i \neq j$ . Let  $c = (c_1, \dots, c_p)$  be proper. The *interpretation of  $\mathbf{G}$  through  $\mathcal{I}$  with parameters  $c$* , denoted by  $\mathcal{I}(\mathbf{G}, c)$ , is the  $\tau$ -structure whose universe is

$$\{a \in G^k : \phi_U(a, c)\}$$

and whose interpretation for  $R_i$  is

$$\{(a_1, \dots, a_r) \in (G^k)^r : \phi_U(a_1, c) \wedge \dots \wedge \phi_U(a_r, c) \wedge \phi_{R_i}(a_1, \dots, a_r, c)\}.$$

If each formula in  $\mathcal{I}$  is quantifier-free then we say that  $\mathcal{I}$  is a *quantifier-free interpretation*.

**Definition.** Let  $\sigma$  and  $\tau$  be finite relational vocabularies, let  $\mathcal{C}$  be a class of  $\sigma$ -structures and let  $\mathcal{D}$  be a class of  $\tau$ -structures closed under isomorphisms. We say that a first-order (resp. quantifier-free) interpretation  $\mathcal{I}$  with  $p$  parameters of  $\tau$  in  $\sigma$  is a *first-order (resp. quantifier-free) reduction of  $\mathcal{C}$  to  $\mathcal{D}$*  if for every  $\sigma$ -structure  $\mathbf{G}$  with at least  $p$  points the following two equivalences hold:

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

All our first-order reductions will have added structure that we will exploit to show they preserve expressibility in *symmetric* Datalog:

**Definition.** We say that a first-order reduction is *positive* if it satisfies the following conditions:

1.  $\phi_U$  is quantifier-free;
2. for every  $\theta$  in  $\tau$ ,  $\phi_\theta$  is built from atomic formulas and equalities using only the existential quantifier, disjunction and conjunction.

Note that we allow the constants FALSE and TRUE as atomic formulas in our reductions.

<sup>3</sup> Note the important technicality in this statement. When we state that, say, 3HORN-SAT is P-complete under first-order reductions, we are stating that for any set  $\mathcal{C}$  of structures that can be recognized in polynomial-time we have  $\mathcal{C} \leq_{\text{FO}} \text{3HORN-SAT}$ . However, it is always assumed in this context that the structures in  $\mathcal{C}$  are equipped with some linear order and this order is crucially used to establish the P-hardness of 3HORN-SAT. The first-order reductions from  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma)$  mentioned in part 2 of the theorem do not rely on this hypothesis.

2.1. The algebraic reductions

**Lemma 2.2.** Let  $\Gamma_0$  be a finite set of relations on  $C$ , let  $\mu : B \rightarrow C$  be a surjective map. Let  $\Gamma_1 = \{\mu^{-1}(\theta) : \theta \in \Gamma_0\}$ . Then  $\text{CSP}(\Gamma_0) = \text{CSP}(\Gamma_1)$ , in particular there is a positive quantifier-free reduction of  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma_1)$ .

**Proof.** This is straightforward: let  $C'$  denote a subset of  $B$  that maps bijectively to  $C$  via  $\mu$ ; then the substructure of  $\Gamma_1$  induced by  $C'$  is obviously isomorphic to  $\Gamma_0$  via  $\mu$ ; hence  $\Gamma_0$  and  $\Gamma_1$  are homomorphically equivalent hence  $\text{CSP}(\Gamma_0) = \text{CSP}(\Gamma_1)$ . The last statement is trivial.  $\square$

**Lemma 2.3.** Let  $\Gamma_1$  be a finite set of relations on  $B$ , and suppose that  $B$  is a subset of  $A$ . Let  $\Gamma_2 = \{B\} \cup \Gamma_1$  where  $B$  is viewed as a unary relation on  $A$  and the relations in  $\Gamma_1$  are viewed as relations on  $A$ . Then there is a positive quantifier-free reduction of  $\text{CSP}(\Gamma_1)$  to  $\text{CSP}(\Gamma_2)$ .

**Proof.** This is also quite easy: let  $\sigma = \{R_1, \dots, R_s\}$  be the type of the structure  $\Gamma_1$  and let  $\tau = \{R_0, R_1, \dots, R_s\}$  be the type of the structure  $\Gamma_2$ . Let  $\mathbf{G}$  be a structure of type  $\sigma$ , and define the structure  $\mathbf{G}'$  of type  $\tau$  as follows: it has the same universe  $G$  as  $\mathbf{G}$ , and let  $R_0(\mathbf{G}') = G$ , and  $R_i(\mathbf{G}') = R_i(\mathbf{G})$  for all  $1 \leq i \leq s$ . It is obvious that there is a homomorphism of  $\mathbf{G}$  to  $\Gamma_1$  if and only if there is one from  $\mathbf{G}'$  to  $\Gamma_2$ ; furthermore the reduction is clearly quantifier-free and positive.  $\square$

To state our next lemma we require the following notation: if  $\theta$  is a  $k$ -ary relation on  $A^2$ , define a  $2k$ -ary relation on  $A$  by

$$\tilde{\theta} = \{(x_1^1, x_2^1, \dots, x_k^1, x_1^2, x_2^2, \dots, x_k^2) : ((x_1^1, x_2^1), \dots, (x_k^1, x_k^2)) \in \theta\}.$$

**Lemma 2.4.** Let  $\Gamma_2$  be a finite set of relations on  $A^2$ . Let  $\Gamma_3 = \{\tilde{\theta} : \theta \in \Gamma_2\}$ . Then there is a positive quantifier-free reduction of  $\text{CSP}(\Gamma_2)$  to  $\text{CSP}(\Gamma_3)$ .

**Proof.** Let  $\sigma = \{R_1, \dots, R_s\}$  be the type of the structure  $\Gamma_2$  and let  $\tau = \{\tilde{R}_1, \dots, \tilde{R}_s\}$  be the type of the structure  $\Gamma_3$ . The reduction is straightforward: given a structure  $\mathbf{G}$  of type  $\sigma$ , we create two disjoint copies of it,  $\mathbf{G}' = \mathbf{G} \times \{0\} \cup \mathbf{G} \times \{1\}$ , and define for each  $1 \leq i \leq s$

$$R_i(\mathbf{G}') = \{(a_1, 0), (a_1, 1), (a_2, 0), \dots, (a_k, 0), (a_k, 1) : (a_1, \dots, a_k) \in R_i(\mathbf{G})\}.$$

It is not difficult to verify that there is a homomorphism from  $\mathbf{G}$  to  $\Gamma_2$  if and only if there is one from  $\mathbf{G}'$  to  $\Gamma_3$ . It remains to show that the reduction is positive and quantifier-free; it is in fact 2-ary with 2 parameters: let

$$\phi_U(x^1, x^2, y^0, y^1) \equiv (x^2 = y^0) \vee (x^2 = y^1)$$

and if  $\theta \in \sigma$  of arity  $r$  let

$$\phi_{\tilde{\theta}}(x_1^1, x_1^2, \dots, x_{2r}^1, x_{2r}^2, y^0, y^1) \equiv [(x_1^1, x_3^1, \dots, x_{2r-1}^1) \in \theta] \bigwedge_{1 \leq i \leq r} [(x_{2i-1}^1 = x_{2i}^1) \wedge (x_{2i-1}^2 = y^0) \wedge (x_{2i}^2 = y^1)]. \quad \square$$

2.2. The “core to idempotent” reduction

**Lemma 2.5.** Let  $\Gamma_0$  be a finite set of relations on  $A$  such that  $\Gamma_0$  is a core. Let  $\Gamma_1 = \Gamma_0 \cup \{\{a\} : a \in A\}$ . Then there is an  $|A|$ -ary, 2-parameter positive quantifier-free reduction of  $\text{CSP}(\Gamma_1)$  to  $\text{CSP}(\Gamma_0)$ .

**Proof.** Let  $\sigma$  be the signature of  $\Gamma_1$  and let  $\tau$  be the signature of  $\Gamma_0$ . Let  $\mathbf{G}_1$  be a structure of type  $\sigma$ . We construct a  $\tau$ -structure  $\mathbf{G}_0$  as follows: it is a copy of  $\mathbf{G}_1$  from which we remove all relations  $R(\mathbf{G}_1)$  with  $R \in \sigma - \tau$ , together with a disjoint copy of  $\Gamma_0$ . To this we add the following: if an element  $g$  of  $\mathbf{G}_1$  is constrained to value  $a$ , then for each  $\theta \in \tau$  we add to  $\theta(\mathbf{G}_0)$  all tuples obtained from a tuple of  $\theta(\Gamma_0)$  by replacing all occurrences of  $a$  by  $g$ . Then  $\mathbf{G}_1$  admits a homomorphism to  $\Gamma_1$  if and only if  $\mathbf{G}_0$  admits a homomorphism to  $\Gamma_0$ : indeed one direction is immediate. Suppose now that there is a homomorphism  $f : \mathbf{G}_0 \rightarrow \Gamma_0$ ; we need to prove that if an element  $g \in G$  is constrained to  $\{a\}$  then  $f(g) = a$ . Because  $\Gamma_0$  is a core, the restriction of  $f$  to the copy of  $\Gamma_0$  in  $\mathbf{G}_0$  is an isomorphism, and hence we may find an automorphism  $\alpha$  of  $\Gamma_0$  such that  $\alpha \circ f$  is the identity when restricted to  $A$ . Consider the substructure of  $\mathbf{G}_0$  induced by the copy of  $\Gamma_0$ , minus the element  $a$  plus the element  $g$ : by construction it is isomorphic to the core  $\Gamma_0$ , and the restriction of  $\alpha \circ f$  to it has all elements of  $A$  in its image except possibly  $a$ : this means that  $f(g) = a$ .

We now proceed to show that this reduction is indeed described by an  $|A|$ -ary, 2-parameter positive quantifier-free interpretation of  $\tau$  in  $\sigma$ . Let  $h = |A|$ . The universe of  $\mathbf{G}_0$  is defined as the diagonal of  $G^h$  (the copy of  $\mathbf{G}_1$ ) together with all tuples of the form  $(y^0, \dots, y^0, y^1, y^0, \dots, y^0)$  where  $y^1$  is in the  $a$ -th position (the copy of  $\Gamma_0$ ).

Let  $\phi_U(x^1, \dots, x^h, y^0, y^1)$  be the formula  $\phi_U^D \vee \phi_U^Y$  where

$$\phi_U^D(x^1, \dots, x^h, y^0, y^1) \equiv \bigwedge_{i \neq j} (x^i = x^j)$$

and

$$\phi_U^Y(x^1, \dots, x^h, y^0, y^1) \equiv \bigvee_{a \in A} \left[ \bigwedge_{j \neq a} (x^j = y^0) \wedge (x^a = y^1) \right].$$



Let  $\theta$  be in  $\tau$ , of arity  $r$ . We define  $\phi_\theta(x_1, \dots, x_r, y^0, y^1)$  as follows: it is  $\phi_\theta^D \vee \phi_\theta^Y \vee \phi_\theta^M$  where

$$\phi_\theta^D(x_1, \dots, x_r, y^0, y^1) \equiv \left[ \bigwedge_{1 \leq i \leq r} \phi_{\theta_i}^D(x_i^1, \dots, x_i^h, y^0, y^1) \right] \wedge (x_1^1, \dots, x_r^1) \in \theta$$

and

$$\phi_\theta^Y(x_1, \dots, x_r, y^0, y^1) \equiv \bigvee_{(a_1, \dots, a_r) \in \theta(\Gamma_0)} \left[ \bigwedge_{1 \leq i \leq r} \bigwedge_{j \neq a_i} (x_i^{a_i} = y^1) \wedge (x_i^j = y^0) \right],$$

and

$$\begin{aligned} \phi_\theta^M(x_1, \dots, x_r, y^0, y^1) \equiv & \bigvee_{a \in A} \bigvee_{(a_1, \dots, a_r) \in \theta(\Gamma_0)} \left[ \bigwedge_{i: a_i = a} \phi_{\theta_i}^D(x_i^1, \dots, x_i^h, y^0, y^1) \wedge (x_i^1 \in \{a\}) \right] \\ & \wedge \left[ \bigwedge_{i: a_i \neq a} \left( \bigwedge_{j \neq a_i} (x_i^{a_i} = y^1) \wedge (x_i^j = y^0) \right) \right]. \quad \square \end{aligned}$$

### 2.3. The reductions for “inferred” constraints

Let  $\Gamma$  be a set of relations and suppose that  $\text{Pol}(\Gamma) \subseteq \text{Pol}(\Gamma')$ . We shall require the following combinatorial descriptions of the relations  $\theta \in \Gamma'$  in terms of those in  $\Gamma$  (see e.g. [9]).

**Lemma 2.6.** *Let  $\Gamma$  and  $\Gamma'$  be finite sets of relations on  $A$ . Then the following conditions are equivalent:*

1.  $\text{Pol}(\Gamma) \subseteq \text{Pol}(\Gamma')$ ;
2. for every  $\theta \in \Gamma'$  of arity  $k$  there exists a (primitive positive) formula

$$\phi(x_1, \dots, x_k) \equiv \exists y_1, \dots, \exists y_m \psi(x_1, \dots, x_k, y_1, \dots, y_m)$$

where  $\psi$  is a conjunction of atomic formulas with relations in  $\Gamma \cup \{=\}$  such that  $(a_1, \dots, a_k) \in \theta$  if and only if  $\phi(a_1, \dots, a_k)$  holds;

3. there exists a finite sequence  $\Gamma = \Gamma_0, \dots, \Gamma_s = \Gamma'$  such that each set of relations  $\Gamma_i$  is obtained from the preceding one by one of the following operations:
  - (a) removing a relation,
  - (b) adding a relation obtained by permuting the variables of a relation,
  - (c) adding the intersection of two relations of the same arity,
  - (d) adding the product of two relations,
  - (e) adding a relation obtained by projecting an  $n$ -ary relation to its first  $n - 1$  variables,
  - (f) adding the equality relation.

We now prove that the constructions (a)–(e) described above induce positive first-order reductions. The last construction, adding the equality, although not first-order, does preserve expressibility in all restrictions of Datalog we use, see [13]. See also [24] Lemma 3.1 for the proof that all 6 are indeed reductions.

**Lemma 2.7.** *Let  $\Gamma'$  be a finite set of relations on  $A$ , let  $\theta \in \Gamma'$  and let  $\Gamma = \Gamma' \setminus \{\theta\}$ . Then there is a positive, 1-ary 0-parameter quantifier-free reduction of  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma')$ .*

**Proof.** This is obvious: given a structure  $\mathbf{G}$  let  $\mathbf{G}'$  be the same as  $\mathbf{G}$  and further set  $\theta(\mathbf{G}') = \emptyset$ .  $\square$

**Lemma 2.8.** *Let  $\Gamma'$  be a finite set of relations on  $A$ , let  $\theta \in \Gamma'$  be  $r$ -ary. Let  $\Gamma = \Gamma' \cup \{\theta_1\}$  where  $\theta_1 = \{(x_{\pi(1)}, \dots, x_{\pi(r)}) : (x_1, \dots, x_r) \in \theta\}$  for some permutation  $\pi$  of  $\{1, \dots, r\}$ . Then there is a 1-ary, 0-parameter positive quantifier-free reduction of  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma')$ .*

**Proof.** Let  $\sigma$  and  $\tau$  be the signatures of the structures  $\mathbf{\Gamma}'$  and  $\mathbf{\Gamma}$  respectively. Given a  $\sigma$ -structure  $\mathbf{G}$ , let  $\mathbf{G}'$  be the  $\tau$ -structure obtained from  $\mathbf{G}$  by removing  $\theta_1$  but adding to  $\theta(\mathbf{G}')$  all appropriate permutations of tuples in  $\theta_1(\mathbf{G})$ . Formally, the interpretation is defined as follows:  $\phi_U \equiv \text{TRUE}$  and for every  $k$ -ary  $\rho \neq \theta$  in  $\sigma$ , let  $\phi_\rho \equiv (x_1, \dots, x_k) \in \rho$  and define

$$\phi_\theta(x_1, \dots, x_r) \equiv (x_1, \dots, x_r) \in \theta \vee (x_{\pi(1)}, \dots, x_{\pi(r)}) \in \theta_1. \quad \square$$

**Lemma 2.9.** *Let  $\Gamma'$  be a finite set of relations on  $A$ , let  $\alpha, \beta \in \Gamma'$  be  $r$ -ary and let  $\gamma = \alpha \cap \beta$ . Let  $\Gamma = \Gamma' \cup \{\gamma\}$ . Then there is a 1-ary, 0-parameter positive quantifier-free reduction of  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma')$ .*

**Proof.** Let  $\sigma$  and  $\tau$  be the signatures of the structures  $\Gamma'$  and  $\Gamma$  respectively. Given a  $\sigma$ -structure  $\mathbf{G}$ , let  $\mathbf{G}'$  be the  $\tau$ -structure obtained from  $\mathbf{G}$  by removing  $\gamma$  but adding to  $\alpha(\mathbf{G}')$  and  $\beta(\mathbf{G}')$  all tuples in  $\gamma(\mathbf{G})$ . Formally the interpretation is defined as follows:  $\phi_U \equiv \text{TRUE}$  and for every  $k$ -ary  $\rho \neq \alpha, \beta$  in  $\sigma$ , let  $\phi_\rho \equiv (x_1, \dots, x_k) \in \rho$  and let

$$\phi_\alpha(x_1, \dots, x_r) \equiv (x_1, \dots, x_r) \in \alpha \bigvee (x_1, \dots, x_r) \in \gamma$$

and

$$\phi_\beta(x_1, \dots, x_r) \equiv (x_1, \dots, x_r) \in \beta \bigvee (x_1, \dots, x_r) \in \gamma. \quad \square$$

**Lemma 2.10.** Let  $\Gamma'$  be a finite set of relations on  $A$ , let  $\alpha, \beta \in \Gamma'$  be  $k$ - and  $r$ -ary respectively. Let  $\gamma = \{(x_1, \dots, x_k, y_1, \dots, y_r) : \bar{x} \in \alpha, \bar{y} \in \beta\}$ , and let  $\Gamma = \Gamma' \cup \{\gamma\}$ . Then there is a 1-ary, 0-parameter positive first-order reduction of  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma')$ .

**Proof.** Let  $\sigma$  and  $\tau$  be the signatures of the structures  $\Gamma'$  and  $\Gamma$  respectively. Given a  $\sigma$ -structure  $\mathbf{G}$ , let  $\mathbf{G}'$  be the  $\tau$ -structure obtained from  $\mathbf{G}$  by removing  $\gamma$  but adding to  $\alpha(\mathbf{G}')$  and  $\beta(\mathbf{G}')$  the appropriate projection of all tuples in  $\gamma(\mathbf{G})$ . Formally the interpretation is defined as follows:  $\phi_U \equiv \text{TRUE}$  and for every  $k$ -ary  $\rho \neq \alpha, \beta$  in  $\sigma$ , let  $\phi_\rho \equiv (x_1, \dots, x_k) \in \rho$ , and

$$\phi_\alpha(x_1, \dots, x_k) \equiv (x_1, \dots, x_k) \in \alpha \bigvee \exists z_1, \dots, z_r (x_1, \dots, x_k, z_1, \dots, z_r) \in \gamma$$

and

$$\phi_\beta(x_1, \dots, x_r) \equiv (x_1, \dots, x_r) \in \beta \bigvee \exists z_1, \dots, z_k (z_1, \dots, z_k, x_1, \dots, x_r) \in \gamma. \quad \square$$

**Lemma 2.11.** Let  $\Gamma'$  be a finite set of relations on  $A$ , let  $\theta \in \Gamma'$  be  $k$ -ary and let  $\theta_1 = \{(x_1, \dots, x_{k-1}) : \exists x_k \bar{x} \in \theta\}$ . Let  $\Gamma = \Gamma' \cup \{\theta_1\}$ . Then there is a  $(k+1)$ -ary, 2-parameter positive quantifier-free reduction of  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma')$ .

**Proof.** Let  $\sigma$  and  $\tau$  be the signatures of the structures  $\Gamma'$  and  $\Gamma$  respectively. Given a  $\sigma$ -structure  $\mathbf{G}$ , let  $\mathbf{G}'$  be the  $\tau$ -structure obtained from  $\mathbf{G}$  by removing  $\theta_1$ , and adding to its universe, for each tuple  $t = (x_1, \dots, x_{k-1})$  of  $\theta_1(\mathbf{G})$  a new element  $x_t$ , and obtaining  $\theta(\mathbf{G}')$  as the union of  $\theta(\mathbf{G})$  and all tuples  $(x_1, \dots, x_{k-1}, x_t)$ .

Formally the interpretation is defined as follows. Define  $\phi_U(x^1, \dots, x^{k+1}, y^0, y^1)$  to be the formula  $\phi_U^D \vee \phi_U^E$  where

$$\phi_U^D(x^1, \dots, x^{k+1}, y^0, y^1) \equiv \bigwedge_{i \neq j} (x^i = x^j)$$

and

$$\phi_U^E(x^1, \dots, x^{k+1}, y^0, y^1) \equiv [(x^k = y^0) \wedge (x^{k+1} = y^1)].$$

Hence the universe of  $\mathbf{G}'$  is defined as the diagonal of  $G^{k+1}$  (the copy of  $\mathbf{G}$ ) together with all tuples of the form  $(x^1, \dots, x^{k-1}, y^0, y^1)$ .

For each  $r$ -ary  $\mu \in \sigma$ , if  $\mu \neq \theta$  then

$$\phi_\mu(x_1, \dots, x_r, y^0, y^1) \equiv \left[ \bigwedge_{1 \leq i \leq r} \phi_U^D(x_i^1, \dots, x_i^{k+1}, y^0, y^1) \right] \wedge (x_1^1, \dots, x_r^1) \in \mu.$$

Finally define

$$\begin{aligned} \phi_\theta(x_1, \dots, x_k, y^0, y^1) &\equiv \bigwedge_{1 \leq i \leq k-1} \phi_U^D(x_i^1, \dots, x_i^{k+1}, y^0, y^1) \bigwedge (x_1^1, \dots, x_{k-1}^1) \in \theta_1 \bigwedge \\ &\bigwedge_{1 \leq j \leq k-1} (x_k^j = x_j^1) \bigwedge (x_k^k = y^0) \bigwedge (x_k^{k+1} = y^1). \quad \square \end{aligned}$$

#### 2.4. Preserving datalog and its restrictions

In what follows we shall require the following definition (see [11]). Let  $\sigma$  be a vocabulary, and let  $S_1, \dots, S_l$  be relation symbols. Consider a second-order formula

$$\Psi \equiv \exists S_1 \dots S_l \forall v_1 \dots v_m \psi$$

where  $\psi$  is a quantifier-free first-order formula over the vocabulary  $\sigma \cup \{S_1, \dots, S_l\}$ : assume that  $\psi$  is in CNF. In particular we may write each conjunct of  $\psi$  as a disjunct of the form

$$\phi \equiv \phi \bigvee \Sigma$$

where  $\phi$  is a disjunct of atomic or negated atomic formulas over  $\sigma$  (we also allow equalities) and  $\Sigma$  is a disjunct of atomic or negated atomic formulas involving only the  $S_i$  (we allow  $\Sigma$  to be “empty”, i.e. FALSE.)

We say that  $\Psi$  is *restricted Krom* if each  $\Sigma$  contains at most one positive occurrence of some  $S_i$  and at most one negative occurrence of some  $S_j$ , i.e. it is of the form (i) “empty”, (ii)  $\bar{x} \in S_i$ , (iii)  $\neg(\bar{x} \in S_j)$  or (iv)  $(\bar{x} \in S_i) \vee \neg(\bar{y} \in S_j)$ . We say that  $\Psi$  is *monotone* if every disjunct of every  $\phi$  is negated, i.e. each  $\phi$  is of the form

$$\phi \equiv \neg(\bar{x} \in \theta) \vee \dots$$



where the  $\theta$  are relational symbols in  $\sigma$ . We say that  $\Psi$  is *symmetric* if, whenever the disjunct

$$\Phi \vee (\bar{x} \in S_i) \vee \neg(\bar{y} \in S_j)$$

appears then its “symmetric” also appears as a conjunct, namely

$$\Phi \vee (\bar{y} \in S_j) \vee \neg(\bar{x} \in S_i).$$

We say that a class of  $\sigma$ -structures  $\mathcal{C}$  is *definable in restricted, monotone, symmetric Krom SNP* if there exists a formula of this form whose models are precisely the members of  $\mathcal{C}$ . We say that  $\mathcal{C}$  is *definable in restricted, monotone, symmetric Krom SNP with equalities* if the same holds, but the formula is over the vocabulary  $\sigma \cup \{=\}$ , i.e. we allow some of the disjuncts in  $\phi$  to be of the form  $\neg(x = y)$  or  $(x = y)$  (of course equality is interpreted in the natural way in each structure).

The following result is an adaptation of Lemma 1 in [11], with elements of the proof of Lemma 3 of [16]. We denote as  $\text{Datalog}(\neq)$  the extension of Datalog in which we allow equalities and disequalities as EDB’s (see [21].) A class  $\mathcal{C}$  of structures is *homomorphism closed* if  $\mathbf{G}' \in \mathcal{C}$  whenever  $\mathbf{G} \in \mathcal{C}$  and  $\mathbf{G}$  admits a homomorphism to  $\mathbf{G}'$ .

**Lemma 2.12.** *Let  $\mathcal{C}$  be a collection of  $\sigma$ -structures. Then conditions (1) and (2) are equivalent, as are (3) and (4):*

1.  $\mathcal{C}$  is definable in linear (symmetric) Datalog;
2.  $\neg\mathcal{C}$  is definable in restricted, monotone, (symmetric) Krom SNP;
3.  $\mathcal{C}$  is definable in linear (symmetric) Datalog( $\neq$ );
4.  $\neg\mathcal{C}$  is definable in restricted, monotone, (symmetric) Krom SNP with equalities.

Furthermore, if  $\mathcal{C}$  is homomorphism closed, then all the above conditions are equivalent.

**Proof.** The equivalence of (1) and (2) for linear Datalog can be found in Lemma 1 of [11]. In fact, inspection of the proof there shows that it also proves the equivalence of (3) and (4), and that symmetry is preserved. In fact, monotonicity is also preserved, i.e. if every atomic formula (except the equalities) appears negated in the Krom formula, then by Dalmau’s construction they will appear positively in the Datalog program.

We must take care of the presence of the constants *TRUE* and *FALSE*, but this is not difficult: we are dealing with conjuncts of the form

$$\bigvee_Q \neg\phi_q$$

so if some  $\phi_q = \text{TRUE}$ , we can simply remove it, and if some  $\phi_q = \text{FALSE}$ , then we may simply remove the whole conjunct that contains it.

It is clear that (1) and (2) imply (3) and (4), so it remains to show the converse, i.e. we show that if  $\mathcal{C}$  is homomorphism closed and is definable in linear (symmetric) Datalog( $\neq$ ) then it is definable in linear (symmetric) Datalog. Let  $P$  be a Datalog( $\neq$ ) program for  $\mathcal{C}$ ; we may assume that it contains no equalities, and that all variables appearing in the scope of an IDB are distinct (see Lemma 3 of [16]: it is not hard to verify that if  $P$  is linear or symmetric, then so is the resulting program of the transformation given there). We may also clearly remove any rule which contains some disequality of the form  $x \neq x$ . Now let  $P'$  be the program obtained from  $P$  by simply deleting from every rule all disequalities; clearly it is linear (symmetric) if  $P$  is. It is proved in Lemma 3 [16] that  $P'$  is actually a program for  $\mathcal{C}$ .  $\square$

**Remark.** As we noted in the proof of the previous lemma, we may assume our Datalog programs do not contain equalities; consequently, by the construction in [11], (see also [13]), we obtain that if a class of structures is definable in monotone, restricted, (symmetric), Krom SNP, then we may assume that in the defining Krom formula, equalities are never used.

**Proposition 2.13.** *Let  $\mathcal{C}$  be a class of structures of signature  $\sigma$  and let  $\mathcal{D}$  be a class of structures of signature  $\tau$ . Let  $\mathcal{I} = (\phi_U, \phi_\theta (\theta \in \tau))$  be a positive first-order reduction with  $p$  parameters from  $\mathcal{C}$  to  $\mathcal{D}$ . If  $\mathcal{D}$  is definable in monotone, restricted, (symmetric), Krom SNP, then the set of structures  $\mathcal{C}^+$  consisting of all  $\sigma$ -structures with less than  $p$  points together with all structures in  $\mathcal{C}$  with at least  $p$  points is definable in monotone, restricted, (resp. symmetric), Krom SNP with equalities.*

**Proof.** Suppose that  $\mathcal{D}$  is definable in monotone, restricted, (symmetric), Krom SNP. Then there is a formula  $\Psi$  defining it, of the following form:

$$\Psi \equiv \exists S_1 \dots S_l \forall v_1 \dots v_m \psi$$

where  $\psi$  is

$$\psi \equiv \bigwedge_T (\phi_t \vee \Sigma_t)$$

where for each  $t \in T$ ,  $\phi_t$  is a disjunct of negated atomic formulas over  $\tau$ , and each  $\Sigma_t$  is a disjunct of at most one positive and one negated atomic formula with symbols in  $\{S_1, \dots, S_l\}$ . As we noted earlier, we may assume that all atomic formulas in the  $\phi$ ’s do not use equality.

Let  $\mathcal{I}$  be our  $k$ -ary reduction with  $p$  parameters from  $\mathcal{C}$  to  $\mathcal{D}$ , with associated formulas  $\phi_U, \phi_\theta$  for each  $\theta \in \tau$ . Consider the following procedure (inspired by Lemma 1 of [3]): our new formula  $\Psi'$  will have first-order variables  $v_1^i, \dots, v_k^i$  for each

$1 \leq i \leq m$  and also  $y^1, \dots, y^p$ , and variables  $u^1, \dots, u^s$ : we assume that all quantified variables in the formulas  $\phi_\theta$  of our reduction appear in this list, and furthermore we assume that no variable  $u^i$  appears in more than one  $\phi_\theta$ . It will also have second-order variables  $S'_1, \dots, S'_l$ .

Let  $\theta$  be a relational symbol from  $\tau$ , and let  $\delta \equiv (v^{i_1}, \dots, v^{i_r}) \in \theta$  be an atomic formula with symbol  $\theta$  and variables among  $v^1, \dots, v^m$ . We define a  $\sigma$ -formula  $\mathcal{I}(\delta)$  as follows:

– if  $\phi_\theta$  is quantifier-free, let

$$\mathcal{I}(\delta) \equiv \phi_\theta(v_1^{i_1}, \dots, v_k^{i_1}, \dots, v_1^{i_r}, \dots, v_k^{i_r}, y^1, \dots, y^p);$$

– if  $\phi_\theta \equiv \exists u^{i_1} \dots u^{i_w} \mu_\theta$  where  $\mu_\theta$  is quantifier-free, let

$$\mathcal{I}(\delta) \equiv \mu_\theta(v_1^{i_1}, \dots, v_k^{i_1}, \dots, v_1^{i_r}, \dots, v_k^{i_r}, u^{i_1}, \dots, u^{i_w}, y^1, \dots, y^p).$$

For each  $\phi_t$  let  $\mathcal{I}(\phi_t)$  denote the result of the above substitutions on the disjuncts of  $\phi_t$ .

Let  $S_i$  be a second-order variable. If  $\delta \equiv (v^{i_1}, \dots, v^{i_r}) \in S_i$  define the  $\sigma$ -formula

$$\mathcal{I}(\delta) \equiv (v_1^{i_1}, \dots, v_k^{i_1}, \dots, v_1^{i_r}, \dots, v_k^{i_r}) \in S'_i;$$

for each  $t$  let  $\Sigma'_t$  denote the result of this substitution on the disjuncts of  $\Sigma_t$ .

Furthermore, define the following  $\sigma$ -formulas: let

$$Y \equiv \bigvee_{1 \leq i \neq j \leq p} y_i = y_j;$$

and let

$$Z \equiv \bigvee_{1 \leq i \leq m} \neg(\phi_U(v_1^i, \dots, v_k^i, y^1, \dots, y^p)).$$

Now we define  $\Psi'$  as follows:

$$\Psi' \equiv \exists S'_1 \dots S'_l \forall v_1^1 \dots v_k^1, \dots, v_1^m \dots v_k^m \forall y^1 \dots y^p \forall u^1 \dots u^s \psi'$$

where  $\psi'$  is

$$\psi' \equiv \bigwedge_T (\mathcal{I}(\phi_t) \vee Y \vee Z \vee \Sigma'_t).$$

In the following 3 claims, we shall prove that this formula has the desired form and that it precisely captures the structures in  $\mathcal{C}^+$ ; from the presence of formula  $Y$  it is obvious that every  $\sigma$ -structure with less than  $p$  points satisfies  $\Psi'$ , hence it will suffice to prove that for structures with at least  $p$  points, it is precisely those in  $\mathcal{C}$  that satisfy it. In the following we will abuse notation slightly and use the names of variables for values assigned to them.

**Claim 1.**  $\Psi'$  is restricted, monotone Krom with equalities. It is symmetric if  $\Psi$  is.

**Proof of Claim 1.** Fix some  $t \in T$ , and suppose that

$$\phi_t \equiv \neg\delta_1 \vee \dots \vee \neg\delta_q$$

where the  $\delta_i$  are atomic formulas. By the hypothesis on our reductions and by construction, each  $\mathcal{I}(\delta_i)$  is a positive Boolean combination of atomic formulas (atomic formulas some of which may be equalities). Writing each  $\mathcal{I}(\delta_i)$  in DNF and applying the negation, one easily sees that  $\mathcal{I}(\phi_t)$  can be expressed as a positive Boolean combination of negated atomic formulas. Hence we may easily rewrite the conjunct  $\mathcal{I}(\phi_t) \vee Y \vee Z \vee \Sigma'_t$  in the form

$$\bigwedge_K \left[ \left( \bigvee_L \varepsilon_{kl} \right) \vee Y \vee Z \vee \Sigma'_t \right]$$

where the  $\varepsilon_{kl}$  are negated atomic formulas. Now it is clear that our formula is monotone, restricted Krom with equalities.

For symmetry, notice that any conjunct of  $\Psi'$

$$\left( \bigvee_L \varepsilon_{kl} \right) \vee Y \vee Z \vee \Sigma'_t$$

is obtained from some conjunct

$$\phi_t \vee \Sigma_t$$

of  $\Psi$ , which appears only if its symmetric  $\phi_t \vee \overline{\Sigma_t}$  appears as a conjunct in  $\Psi$ , which means that the conjunct

$$\left( \bigvee_L \varepsilon_{kl} \right) \vee Y \vee Z \vee \overline{\Sigma'_t}$$

appears in  $\Psi'$  (obviously the symmetric of  $\Sigma'_t$  is  $\Sigma'_u$  if  $\overline{\Sigma_t} = \Sigma_u$ ).  $\square$

**Claim 2.** Let  $\mathbf{G}$  be a  $\sigma$ -structure with at least  $p$  points, and let  $c$  be proper. Suppose that  $\mathcal{I}(G, c)$  satisfies  $\Psi$ . Then  $\mathbf{G}$  satisfies  $\Psi'$ .

**Proof of Claim 2.** By definition of our reductions the choice of  $c$  is immaterial, i.e.  $\mathcal{I}(G, c)$  satisfies  $\Psi$  for any proper  $c$ .

We must find relations  $S'_i$  on  $G$ : take the obvious choice, namely, if  $S_i$  is  $r$ -ary, let  $S'_i = S_i$  viewed as a  $kr$ -ary relation on  $G$  (rather than an  $r$ -ary relation on  $\mathcal{I}(G, c) \subseteq G^k$ ), i.e.

$$(x_1^1, \dots, x_k^1, \dots, x_1^r, \dots, x_k^r) \in S'_i$$

precisely if

$$((x_1^1, \dots, x_k^1), \dots, (x_1^r, \dots, x_k^r)) \in S_i.$$

Now choose any  $v_1^1, \dots, v_k^1, \dots, v_1^m, \dots, v_k^m, y^1, \dots, y^p, u^1, \dots, u^s$  in  $G$ , and pick any conjunct

$$\mathcal{I}(\phi_t) \vee Y \vee Z \vee \Sigma'_t$$

of  $\Phi'$ : we must show that for this choice of values this conjunct is satisfied. Suppose then that none of  $Y, Z$  nor  $\Sigma'_t$  is satisfied: we must show that  $\mathcal{I}(\phi_t)$  holds.

We make a couple of useful observations:

1. Since our values do not satisfy  $Y$ , we have that  $c = (y^1, \dots, y^p)$  is proper. Hence  $\mathcal{I}(G, c)$  satisfies  $\Psi$  for  $c = (y^1, \dots, y^p)$ .
2. Since our values do not satisfy  $Z$ , it means that each tuple  $v^i = (v_1^i, \dots, v_k^i)$  is in the universe of  $\mathcal{I}(G, c)$  (i.e. satisfies  $\phi_U$ ).

Since  $\mathcal{I}(G, c)$  satisfies the conjunct

$$\phi_t \vee \Sigma_t$$

with  $v^1, \dots, v^m$ , by definition of the  $S'_i$  and  $\Sigma'_t$  it follows immediately that  $\Sigma_t$  is *not* satisfied by  $v^1, \dots, v^m$ , hence  $\mathcal{I}(G, c)$  satisfies  $\phi_t$  (with  $v^1, \dots, v^m$ ).

There is some index set  $Q$  such that

$$\phi_t \equiv \bigvee_Q \neg(\gamma_q)$$

where each  $\gamma_q$  is an atomic formula with relational symbol in  $\tau$ . Since  $\mathcal{I}(G, c)$  satisfies it there exists some  $q \in Q$  such that  $\gamma_q$  does not hold. Let

$$\gamma_q \equiv (v^{i_1}, \dots, v^{i_r}) \in \theta$$

and suppose first that  $\phi_\theta$  is quantifier-free. By (2) and the definition of interpretation it is immediate that  $\mathbf{G}$  does not satisfy the corresponding occurrence of  $\phi_\theta$  in  $\mathcal{I}(\phi_t)$ . Similarly, if  $\phi_\theta$  is not quantifier-free, say

$$\phi_\theta \equiv \exists u^{i_1} \dots u^{i_w} \mu_\theta$$

is not satisfied, which means that for the values  $u^1, \dots, u^s$  we have,  $\mu_\theta$  is not satisfied, and hence  $\mathbf{G}$  does not satisfy the corresponding occurrence in  $\mathcal{I}(\phi_t)$ . Hence we're done.  $\square$

**Claim 3.** Let  $\mathbf{G}$  be a  $\sigma$ -structure with  $p$  distinct points. If  $\mathbf{G}$  satisfies  $\Psi'$  then  $\mathcal{I}(G, c)$  satisfies  $\Psi$  for any proper  $c$ .

**Proof of Claim 3.** Choose some proper  $c = (z_1, \dots, z_p)$ . For each  $1 \leq i \leq l$  let  $S_i$  be the  $r$ -ary relation that consists of those tuples of  $S'_i$  that are in the universe of  $\mathcal{I}(G, c)$ , i.e.

$$((x_1^1, \dots, x_k^1), \dots, (x_1^r, \dots, x_k^r)) \in S_i$$

if and only if

$$(x_1^1, \dots, x_k^1, \dots, x_1^r, \dots, x_k^r) \in S'_i$$

and  $\phi_U(x_1^i, \dots, x_k^i, y^1, \dots, y^p)$  holds for all  $1 \leq i \leq m$ .

Choose some elements  $v^i = (v_1^i, \dots, v_k^i)$  in the universe of  $\mathcal{I}(G, c)$  (i.e. they all satisfy  $\phi_U$ ), and choose some conjunct  $\phi_t \vee \Sigma_t$  of  $\Psi$ . Suppose that  $\Sigma_t$  does not hold: we must show that  $\phi_t$  does (for our choice of  $v^i$ 's). Now by definition of the  $S_i$  and  $\Sigma'_t$  it is immediate that  $\Sigma'_t$  isn't satisfied either. Hence  $Y \vee Z \vee \mathcal{I}(\phi_t)$  holds for any choice of  $u^1, \dots, u^s$ . However, by choice of the  $z_i$ , all distinct, and the  $v^i$ , we have that neither  $Y$  nor  $Z$  is satisfied, which means that  $\mathcal{I}(\phi_t)$  holds for any choice of  $u^1, \dots, u^s$ . We will conclude that  $\phi_t$  holds in  $\mathcal{I}(G, c)$ : in fact the argument is quite similar to the one used in the last claim. As above let

$$\phi_t \equiv \bigvee_Q \neg(\gamma_q)$$

where each  $\gamma_q$  is an atomic formula with relational symbol in  $\tau$ . Suppose for a contradiction that every  $\gamma_q$  holds in  $\mathcal{I}(G, c)$ . Choose some  $q \in Q$  and suppose first that

$$\gamma_q \equiv (v^{i_1}, \dots, v^{i_r}) \in \theta$$

where  $\phi_\theta$  is quantifier-free. As we argued in Claim 2 it is immediate that  $\mathbf{G}$  satisfies the corresponding occurrence of  $\phi_\theta$  in  $\mathcal{L}(\phi_t)$ . Now suppose that  $\phi_\theta$  is not quantifier-free, say

$$\phi_\theta \equiv \exists u^{i_1} \dots u^{i_w} \mu_\theta$$

this means that we can find  $u^{i_1} \dots u^{i_w}$  in  $\mathbf{G}$  that satisfy  $\mu_\theta$ ; since no variable  $u^i$  appears in more than one of the  $\phi_\theta$ , it means we may find an assignment of values  $u^1, \dots, u^s$  such that every atomic formula appearing in  $\mathcal{L}(\phi_t)$  is satisfied, i.e. so that  $\mathbf{G}$  does not satisfy  $\mathcal{L}(\phi_t)$ , contradicting our hypothesis.  $\square$

From the 3 claims we can now conclude what we wanted: if  $G$  contains at least  $p$  distinct points,  $G \in \mathcal{C}$  iff there exists some proper  $c$  such that  $\mathcal{L}(G, c) \in \mathcal{D}$  iff  $\mathcal{L}(G, c)$  satisfies  $\Psi$  iff  $\mathbf{G}$  satisfies  $\Psi'$ .  $\square$

**Corollary 2.14.** *Let  $\mathcal{C}$  be homomorphism closed. Suppose that  $\neg\mathcal{C}$  reduces to  $\neg\mathcal{D}$  via a positive first-order reduction. If  $\mathcal{D}$  is definable in linear (symmetric) Datalog, then  $\mathcal{C}$  is definable in linear (symmetric) Datalog.*

**Proof.** By Lemma 2.12, if  $\mathcal{D}$  is definable in linear (symmetric) Datalog, then  $\neg\mathcal{D}$  is definable in restricted, monotone, (symmetric) Krom SNP. Since  $\neg\mathcal{C}$  reduces to  $\neg\mathcal{D}$  via a positive first-order reduction, by Proposition 2.13 we conclude that for some  $p \geq 0$ , the set consisting of all  $\sigma$ -structures with less than  $p$  points or that are in  $\neg\mathcal{C}$  with at least  $p$  points is definable in restricted, monotone, (symmetric) Krom SNP with equalities. It follows from another application of Lemma 2.12 that the set of all structures in  $\mathcal{C}$  with at least  $p$  points is definable in linear (symmetric) Datalog( $\neq$ ). To the program accepting this set of structures, and for every structure  $G \in \mathcal{C}$  with less than  $p$  points, add the rule

$$P() \leftarrow \phi(x_1, \dots, x_k)$$

where  $P()$  is the goal predicate of the original program,  $x_1, \dots, x_k$  represent the elements of  $G$  and  $\phi$  is the conjunction of all relations that hold in  $\mathbf{G}$ . Obviously the new program will now accept every structure in  $\mathcal{C}$ , and it is easy to see that since  $\mathcal{C}$  is homomorphism closed, no other structure can be accepted. It is also obvious that if the original program is linear or symmetric, so is the new one. Hence  $\mathcal{C}$  is definable in linear (symmetric) Datalog( $\neq$ ); we invoke Lemma 2.12 one last time to conclude that it is definable in linear (symmetric) Datalog.  $\square$

The arguments of the previous two subsections are now sufficient to prove Theorem 2.1.

**Theorem. (2.1).** *Let  $\Gamma$  be a finite set of relations on  $A$  such that  $\mathbf{\Gamma}$  is a core. Let  $\mathbb{A}$  denote the full idempotent reduct of the algebra associated to  $\Gamma$ .*

1. *Let  $\mathbb{C}$  be a finite algebra in  $\mathcal{V}(\mathbb{A})$ , and let  $\Gamma_0$  be a finite set of relations invariant under the basic operations of  $\mathbb{C}$ . Then there exists a logspace many-one reduction of  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma)$ . Furthermore, if  $\neg\text{CSP}(\Gamma)$  is expressible in (linear, symmetric) Datalog, then so is  $\neg\text{CSP}(\Gamma_0)$ .*
2. *If furthermore  $\mathbb{C} \in \text{HS}(\mathbb{A})$  and the relations in  $\Gamma_0$  are irredundant, then the above reduction is first-order without ordering.*

**Proof.** (1) We shall define a sequence of CSP each of which reduces to the next in the proper fashion for our needs. We note that all reductions we use preserve expressibility in plain Datalog by [24]. By hypothesis on the algebra  $\mathbb{C}$  there exist an integer  $m$ , a subalgebra  $\mathbb{B}$  of  $\mathbb{A}^m$  and a surjective homomorphism  $\mu : \mathbb{B} \rightarrow \mathbb{C}$ .

Let  $\Gamma_1 = \{\mu^{-1}(\theta) : \theta \in \Gamma_0\}$ . By Lemma 2.2 there is a (positive) first-order reduction of  $\text{CSP}(\Gamma_0)$  to  $\text{CSP}(\Gamma_1)$  which by Corollary 2.14 preserves expressibility in (linear, symmetric) Datalog. Furthermore, it is easy to see that all relations in  $\Gamma_1$  are invariant under the operations of  $\mathbb{B}$ .

Next define  $\Gamma_2 = \Gamma_1 \cup \{B\}$ , viewed as relations on  $A^m$ . By Lemma 2.3 we have a (positive) first-order reduction of  $\text{CSP}(\Gamma_1)$  to  $\text{CSP}(\Gamma_2)$  which preserves expressibility in (linear, symmetric) Datalog by Corollary 2.14. Furthermore, it is easy to see that all relations in  $\Gamma_2$  are invariant under the operations of  $\mathbb{A}^m$ .

Now define  $\Gamma_3 = \{\theta : \theta \in \Gamma_2\}$  (see definition just before Lemma 2.4). Once again we obtain a (positive) first-order reduction of  $\text{CSP}(\Gamma_2)$  to  $\text{CSP}(\Gamma_3)$  preserving expressibility in restrictions of Datalog by Lemma 2.4 and Corollary 2.14.

It is easy to see that the relations in  $\Gamma_3$  are invariant under the basic operations of  $\mathbb{A}$ . Since these are precisely the idempotent operations that preserve all relations in  $\Gamma$ , we obtain that  $\text{Pol}(\Gamma') \subseteq \text{Pol}(\Gamma_3)$  where  $\Gamma' = \Gamma \cup \{\{a\} : a \in A\}$ . There exists a finite sequence of sets  $\Gamma' = \Lambda_0, \dots, \Lambda_s = \Gamma_3$  such that each set is obtained from the previous by one of the 6 constructions described in Lemma 2.6 (3). By Lemma 2.7 to 2.11 and Corollary 2.14, the first 5 constructions induce first-order reductions that preserve expressibility in the various restrictions of Datalog we need. The 6th reduction is proved to be logspace<sup>4</sup> in [20], and also preserves expressibility by [13] (see also [14]). Hence we have a logspace reduction of  $\text{CSP}(\Gamma_3)$  to  $\text{CSP}(\Gamma')$ , that preserves expressibility.

Finally we invoke Lemma 2.5 to obtain a first-order reduction of  $\text{CSP}(\Gamma')$  to  $\text{CSP}(\Gamma)$  which preserves expressibility by Corollary 2.14.

(2) Now we prove the second statement: notice that in the above argument we would have obtained a first-order reduction had it not been for the reduction associated to adding the equality relation. So we show that if the relations in  $\Gamma_0$

<sup>4</sup> The reduction provided in [20] is logspace because of Reingold's result showing that the symmetric transitive closure of a binary relation can be computed in logspace [26].

are irredundant and we do not use the reduction associated to powers, then we need not use it. Follow the above construction letting  $m = 1$  and thus  $\Gamma_3 = \Gamma_2$ ; it is clear that in the above construction the relations in this set must be irredundant if those in  $\Gamma_0$  are. Now we show that in the sequence of sets  $\Lambda_i$  we never need to add the equality relation. Indeed, by part (2) of Lemma 2.6 for every  $\gamma \in \Gamma_3$  there exists a primitive positive formula defining it in terms of the relations in  $\Gamma'$ . We claim that if  $\gamma$  is irredundant then we need not use equality in our formula. Indeed, suppose that the atom  $x = y$  appears in the formula: if either of the two variables is quantified, say  $x$ , we may simply rename it everywhere in the formula to  $y$  and remove  $\exists x$ . Repeating this, we obtain a formula in which equalities can occur only between free variables, which is impossible since the relation defined is irredundant. It remains to be seen that we can achieve this by using only the 5 other reductions: it is clear that  $\gamma$  is obtained as a projection of a relation defined by a quantifier-free formula, which consists of a conjunction of atomic formulas of the form  $\bar{x} \in \theta_j$  for relations  $\theta_j \in \Gamma$ . So let's say that  $x_1, \dots, x_h$  is a list of all distinct variables appearing in the formula. For each conjunct  $(x_{i_1}, \dots, x_{i_s}) \in \theta_j$  the relation

$$\{(x_1, \dots, x_h) : (x_{i_1}, \dots, x_{i_s}) \in \theta_j\}$$

is obtained by a permutation of the product  $A^{h-s} \times \theta_j$ .<sup>5</sup> Finally, our relation is the intersection of all these relations.  $\square$

### 3. Finite duality and L-hardness

A relational structure  $\Gamma$  has *finite duality* if there exist finitely many structures  $\mathbf{H}_1, \dots, \mathbf{H}_k$  such that a structure  $\mathbf{G}$  belongs to  $\text{CSP}(\Gamma)$  if and only if none of the  $\mathbf{H}_i$  admits a homomorphism to  $\mathbf{G}$ . It is immediate that  $\Gamma$  has finite duality precisely if  $\neg\text{CSP}(\Gamma)$  is definable by a non-recursive Datalog program. In this section, we shall show that if  $\Gamma$  does not have finite duality then  $\text{CSP}(\Gamma)$  is L-hard. As a consequence, we shall obtain an alternate proof of the result of Atserias [3] mentioned at the end of Section 1.2, that states that  $\text{CSP}(\Gamma)$  is first-order definable precisely when  $\Gamma$  has finite duality.

We require a characterisation of relational structures with finite duality from [22]: although the result is stated there for FO-definable structures, it is easy to see that in fact it deals only with structures with finite duality. Consider the signature  $\sigma = \{R_1, \dots, R_r\}$  where  $R_i$  is a relational symbol of arity  $a_i$ . For an integer  $n$  the  $n$ -link of type  $\sigma$  is the  $\sigma$ -structure

$$\mathbf{L}_n = \langle \{0, 1, \dots, n\}; R_1(\mathbf{L}_n), \dots, R_r(\mathbf{L}_n) \rangle,$$

such that  $R_i(\mathbf{L}_n) = \cup_{j=1}^n \{j-1, j\}^{a_i}$  for  $i = 1, \dots, r$ . Intuitively, a link is obtained from a path  $0, 1, \dots, n$  by replacing each edge by the relational structure of type  $\sigma$  on 2 elements whose basic relations are of maximal size.

Let  $\mathbf{A}$  and  $\mathbf{B}$  be two  $\sigma$ -structures. The  $\mathbf{A}$ -th power of  $\mathbf{B}$  is the  $\sigma$ -structure

$$\mathbf{B}^{\mathbf{A}} = \langle \mathbf{B}^{\mathbf{A}}; R_1(\mathbf{B}^{\mathbf{A}}), \dots, R_r(\mathbf{B}^{\mathbf{A}}) \rangle,$$

where  $\mathbf{B}^{\mathbf{A}}$  is the set of all maps from  $A$  to  $B$ , and for  $i = 1, \dots, r$  the relation  $R_i(\mathbf{B}^{\mathbf{A}})$  consists of all tuples  $(f_1, \dots, f_{a_i})$  such that  $(f_1(x_1), \dots, f_{a_i}(x_{a_i})) \in R_i(\mathbf{B})$  whenever  $(x_1, \dots, x_{a_i}) \in R_i(\mathbf{A})$ .

Let  $\pi_1$  and  $\pi_2$  denote the two projections from  $A^2$  to  $A$ .

**Lemma 3.1** ([22]). *Let  $\Gamma$  be a core. Then  $\Gamma$  has finite duality if and only if for some  $n$  there exists a homomorphism  $P : \mathbf{L}_n \rightarrow \Gamma(\Gamma^2)$  such that  $P(0) = \pi_1$  and  $P(n) = \pi_2$ .*

**Theorem 3.2.** *Let  $\Gamma$  be a finite structure. If  $\Gamma$  does not have finite duality then the problem  $\text{CSP}(\Gamma)$  is L-hard under first-order reductions.*

**Proof.** We may assume that  $\Gamma$  is a core. Let  $\sigma = \{R_1, \dots, R_r\}$  be the signature of the structure  $\Gamma$  and let  $\Gamma' = \Gamma \cup \{\{a\} : a \in A\}$ . For  $a \in A$  let  $S_a$  be a relational symbol for  $\{a\}$ , and let  $\sigma' = \{R_1, \dots, R_r\} \cup \{S_a : a \in A\}$  be the signature of the structure  $\Gamma'$ . By Theorem 2.1 (see Lemma 2.5) it suffices to show that  $\text{CSP}(\Gamma')$  is L-hard: for this we construct a first-order reduction of NOT  $st$ -connectivity to  $\text{CSP}(\Gamma')$ . Consider the vocabulary of graphs with two specified vertices,  $\tau = \{E, s, t\}$  where  $E$  is a binary relational symbol and  $s$  and  $t$  are constant symbols. We shall describe a first-order interpretation  $\mathcal{I}$  of  $\sigma'$  into  $\tau$  assigning to each graph  $\mathbf{G}$  with distinguished vertices  $s$  and  $t$  a structure  $\mathbf{K} = \mathcal{I}(\mathbf{G})$  of type  $\sigma'$  such that  $\mathbf{K}$  admits a homomorphism to  $\Gamma'$  precisely when  $s$  and  $t$  are not connected in  $\mathbf{G}$ .

It is clear that the following defines a symmetric relation  $\sim$  on  $\Gamma(\Gamma^2)$ : let  $g \sim h$  if there exists a homomorphism  $F : \mathbf{L}_1 \rightarrow \Gamma(\Gamma^2)$  such that  $F(0) = g$  and  $F(1) = h$ . It is also clear by definition of the links that Lemma 3.1 shows this: for a core  $\Gamma$ ,  $\Gamma$  has finite duality precisely when the projections are connected in the graph defined by  $\sim$ .

We first describe the reduction: the argument that it is indeed first-order will follow. Given a graph  $\mathbf{G}$  with specified vertices  $s$  and  $t$ , we construct a  $\sigma$ -structure  $\mathbf{H}$  which is obtained from  $\mathbf{G}$  by replacing each edge by the link  $\mathbf{L}_1$  (in the same manner that links are obtained from paths). Consider now the product  $\sigma$ -structure  $\mathbf{H} \times \Gamma^2$ , that we transform into the  $\sigma'$ -structure  $\mathbf{K} = \mathcal{I}(\mathbf{G})$  as follows: for each  $a \in A$  we define  $S_a(\mathbf{K})$  to contain all elements  $(g, c, d)$  such that  $(g = s$  and  $c = a)$  OR  $(g = t$  and  $d = a)$ . We first show that the above is indeed a reduction of NOT  $st$ -connectivity to  $\text{CSP}(\Gamma')$ . Suppose there is a homomorphism  $f$  from  $\mathbf{K}$  to  $\Gamma'$ : in particular it is a homomorphism of  $\sigma$ -structures  $f : \mathbf{H} \times \Gamma^2 \rightarrow \Gamma$ , which, by the natural property of products, induces a homomorphism  $F$  from  $\mathbf{H}$  to  $\Gamma^2$ ; by definition of the relations  $S_a(\mathbf{K})$ , it is easy to verify that  $F(s) = \pi_1$  and  $F(t) = \pi_2$ . Indeed, we have  $F(s)(c, d) = f(s, c, d) = c$  and  $F(t)(c, d) = f(t, c, d) = d$ . Since  $\Gamma$  does not have finite duality, there is no path between the projections in the graph defined by  $\sim$ , hence there cannot be a path in  $\mathbf{G}$  from  $s$  to  $t$ . Conversely, suppose there is no such path in  $\mathbf{G}$ . Then define a map  $f$  from  $\mathbf{H} \times \Gamma^2$  to  $\Gamma$  by setting  $f(g, c, d) = c$  if there

<sup>5</sup> Note of course, that the unary relation  $A$  can be added to a CSP with base set  $A$  without affecting its complexity.

is a path in  $\mathbf{G}$  from  $s$  to  $g$  and  $f(g, c, d) = d$  otherwise. This is clearly well-defined, and obviously preserves all the relations  $S_a$ . It is easy to see that  $f$  also preserves all relations  $R_i$ ; indeed, the map  $F : \mathbf{H} \rightarrow \mathbf{\Gamma}^{\Gamma^2}$  induced by  $f$  maps all elements to one of two projections which are “loops” in any power structure.  $\square$

We now complete the proof of the theorem by showing that the reduction is indeed FO.

Pick an integer  $m$  such that  $2^m \geq |A|^2$  and fix an encoding of the elements of  $A^2$  as sequences of length  $m$  over  $\{s, t\}$ .

- The universe of  $\mathcal{L}(\mathbf{G}) = \mathbf{K}$  is the subset of  $G \times G^m$  consisting of all tuples  $(g; x^1, \dots, x^m)$  such that  $g \in G$  and  $(x^1, \dots, x^m)$  is the encoding of an element of  $A^2$ . Notice that this is first-order defined, since we may simply list all possibilities, i.e.

$$[(x^1 = s) \wedge \dots \wedge (x^m = t)] \bigvee \dots$$

- Fix  $1 \leq i \leq r$  and let  $a =$  denote the arity of  $R_i$ ; we define the relation  $R_i(\mathbf{K})$  as the set of all tuples

$$[(g_1; x_1^1, \dots, x_1^m), \dots, (g_a; x_a^1, \dots, x_a^m)]$$

such that

1. there exists  $x, y \in G$  such that  $xEy$  and  $\{g_1, \dots, g_a\} \subseteq \{x, y\}^a$ , and
2.  $(y_1, \dots, y_a) \in R_i(\mathbf{\Gamma}^2)$  where  $y_j$  is the element of  $A^2$  indexed by  $(x_j^1, \dots, x_j^m)$  for every  $j$ .

It is easy to see that both these conditions are describable by a first-order formula: indeed, for the first condition, simply notice we can list all possible subsets of  $\{x, y\}^a$  and so the formula will be of the form

$$\exists x, y (xEy) \wedge \left[ \{g_1 = x \wedge g_2 = x \wedge \dots \wedge g_a = x\} \bigvee \dots \right].$$

- For the unary relations  $S_a$  we proceed as follows: for a fixed  $a \in A$ , define  $S_a(\mathbf{K})$  to be the set of all tuples  $(g; x^1, \dots, x^m)$  such that either

1.  $g = s$  and  $(x^1, \dots, x^m)$  is the index of a tuple of  $A^2$  of the form  $(a, y)$  for some  $y$ , or
2.  $g = t$  and  $(x^1, \dots, x^m)$  is the index of a tuple of  $A^2$  of the form  $(x, a)$  for some  $x$ .

Once again it is easy to see that these are first-order defined as we may simply list all possible indices of tuples that we require. This completes the proof.

Recall that the complexity class non-uniform  $\mathbf{AC}^0$  consists of all languages accepted by polynomial-size constant-depth families of Boolean circuits. It is known that any FO-definable class of structures belongs to this complexity class (see Theorem 6.4 of [25]). As noted in Section 1, the class non-uniform  $\mathbf{AC}^0$  is closed under first-order reductions. Moreover, there are problems in  $L$  which are not in non-uniform  $\mathbf{AC}^0$  (see [17]) and so any problem which is  $L$ -hard under first-order reductions cannot lie in non-uniform  $\mathbf{AC}^0$ . These facts and the last result yield the following:

**Theorem 3.3.** *For any structure  $\mathbf{\Gamma}$ , the following conditions are equivalent:*

1.  $\mathbf{\Gamma}$  has finite duality.
2.  $\text{CSP}(\mathbf{\Gamma})$  is FO-definable.
3.  $\text{CSP}(\mathbf{\Gamma})$  is in non-uniform  $\mathbf{AC}^0$ .

The first two statements were shown as equivalent by Atserias [3] and a more general statement was established by Rossman [27]. Note also that the third equivalence could be strengthened by replacing non-uniform  $\mathbf{AC}^0$  by any class known to exclude  $L$ -hard problems, such as non-uniform  $\mathbf{AC}^0[\mathbf{p}]$  for a prime  $p$ .

#### 4. Main theorems

We present in this section our two main theorems. The first provides sufficient algebraic criteria for the hardness of  $\text{CSP}(\mathbf{\Gamma})$  for a number of natural complexity classes. The second expresses these same lower bounds in descriptive complexity terms.

**Theorem 4.1.** *Let  $\Gamma$  be a finite set of relations on  $A$  such that  $\mathbf{\Gamma}$  is a core, and let  $\mathbb{A} = \mathbb{A}(\mathbf{\Gamma})$ . Then:*

1. If  $\mathcal{V}(\mathbb{A})$  admits the unary type, then  $\text{CSP}(\mathbf{\Gamma})$  is NP-complete under FO reductions.
2. If  $\mathcal{V}(\mathbb{A})$  omits the unary type but admits the affine type, then there exists a prime  $p$  such that  $\text{CSP}(\mathbf{\Gamma})$  is  $\text{Mod}_p L$ -hard under FO reductions.
3. If  $\text{CSP}(\mathbf{\Gamma})$  is not FO, then it is  $L$ -hard under FO reductions.
4. If  $\mathcal{V}(\mathbb{A})$  omits the unary, and semilattice types, but admits the lattice type, then  $\text{CSP}(\mathbf{\Gamma})$  is NL-hard under FO reductions.
5. If  $\mathcal{V}(\mathbb{A})$  omits the unary type, but admits the semilattice type, then  $\text{CSP}(\mathbf{\Gamma})$  is  $P$ -hard under FO reductions.

**Proof.** (3) is the content of Theorem 3.2, and (1) follows from a result of [7]: the reduction there is actually first-order by Theorem 2.1.

It follows from results in [18] that if  $\mathbb{A}$  satisfies the hypothesis of one of (2), (4) or (5) then so does its full idempotent reduct, which we denote by  $\mathbb{B}$ . By Lemma 1.3, if  $\mathbb{B}$  satisfies the hypothesis (i) then there exists a strictly simple algebra  $\mathbb{C} \in \text{HS}(\mathbb{B})$  of type (i).

By Corollary 1.2 this means that, in case (4), one can find an FO reduction of  $\text{CSP}(\mathbf{\Gamma}')$  to  $\text{CSP}(\mathbf{\Gamma})$  where  $\mathbf{\Gamma}' = \{\leq, \{0\}, \{1\}\}$ . There is a straightforward FO-reduction of the directed graph unreachability problem to  $\text{CSP}(\mathbf{\Gamma}')$  and the former problem is NL-complete under first-order reductions [19].



Similarly, in case (5), we find through [Corollary 1.2](#), a first-order reduction of  $\text{CSP}(\Gamma')$  to  $\text{CSP}(\Gamma)$  where  $\Gamma' = \{\theta, \{0\}, \{1\}\}$  where  $\theta = \{(x, y, z) : (y \wedge z) \rightarrow x\}$ . It is again straightforward to show that  $\text{CSP}(\Gamma')$  admits a natural FO reduction of  $\text{HORN-3-SAT}$  which is P-hard under FO-reductions [19].

Finally, in case (2), there exists an Abelian group structure on the base set of  $\mathbb{C}$  such that the 3-ary relation  $\mu = \{(x, y, z) : x+y = z\}$  is invariant under the operations of  $\mathbb{C}$ . Consider the set  $\Gamma'$  that consists of the relation  $\mu$ , the unary relation  $B = \{b\}$  where  $b$  is some non-zero element of  $\mathbb{C}$  such that  $pb = 0$  for some prime  $p$ , and the unary relation  $Z = \{0\}$ . By [Lemma 1.4](#), it will suffice to find a first-order reduction of  $\text{NOT } p\text{-GAP}$  to  $\text{CSP}(\Gamma')$ . The basic idea is rather straightforward: Let  $\mathbf{G}$  be a digraph with specified vertices  $s$  and  $t$ . Recall that (a) the input digraph  $\mathbf{G}$  for the problem  $\text{NOT } p\text{-GAP}$  is equipped with a linear order such that edges are monotone and  $1 = s$  and  $n = t$ ; and (b) we may assume that  $\mathbf{G}$  has no loops. Notice further that we may also invoke the successor relation which is first-order definable from the ordering; for ease of notation in formulas we shall denote the successor of  $i$  simply by  $i + 1$ . We construct a system of equations over the group  $\mathbb{A}$ , that admits a solution precisely if the number of paths from  $s$  to  $t$  is divisible by  $p$ . The system has variables  $x_{i,j}$  for all  $i \leq j$  in  $G$ : we shall set up the equations so that we interpret the value of  $x_{i,j}$  in any solution (which will turn out to be unique) as the number of paths in  $\mathbf{G}$  from  $1$  to  $j$  that go through an edge  $(k, j)$  for some  $k \leq i$ . Here are the equations:

1. for every  $1 \leq j$ , let  $x_{1,j} = b$  if  $(1, j)$  is an edge of  $\mathbf{G}$ , and otherwise let  $x_{1,j} = 0$ .
2. For  $i + 1 < j$ , let  $x_{i,j} + x_{i+1,i+1} = x_{i+1,j}$  if  $(i + 1, j)$  is an edge of  $\mathbf{G}$ , and let  $x_{i,j} = x_{i+1,j}$  otherwise;
3. for every  $i < n$  let  $x_{i,i+1} = x_{i+1,i+1}$ ;
4. finally, also put  $x_{n,n} = 0$ .

It is not hard to convince oneself that the following holds: the unique solution (if it exists) is the following:  $x_{i,j} = mb$  where  $m$  is the number of paths in  $\mathbf{G}$  from  $1$  to  $j$  that go through an edge  $(k, j)$  for some  $k \leq i$ . Thus, the system has a solution if and only if the number of paths from  $1$  to  $n$  (i.e. from  $s$  to  $t$ ) in  $\mathbf{G}$  is divisible by  $p$ .

We now show that the above construction is first-order. Let  $\mathbf{G}$  be a digraph with edge relation  $\theta(G)$  and vertices  $1 = s$  and  $t = n$ . Let  $\tau = \langle \rho, B, Z \rangle$  denote the signature of the target structure of our CSP. We construct a structure  $\mathbf{H}$  of type  $\tau$  that will admit a homomorphism to the target structure if and only if the number of paths from  $s$  to  $t$  in  $\mathbf{G}$  is divisible by  $p$ . It should be clear how to translate back and forth from the relational description to a system of equations: for instance the equation  $x + y = b$  is equivalent to  $(x, y, z) \in \rho$  and  $z \in B$ .

Define the universe of  $\mathbf{H}$  by the following formula:

$$\phi_U(i, j) \equiv i \leq j;$$

in other words, the pair  $(i, j)$  stands for the variable  $x_{i,j}$ . Next, define the unary relations  $B(\mathbf{H})$  and  $Z(\mathbf{H})$  by the following formulas:

$$\phi_B(i, j) \equiv [(i = s) \wedge ((i, j) \in \theta)]$$

and

$$\phi_Z(i, j) \equiv [(i = s) \wedge \neg((i, j) \in \theta)] \vee [(i = t) \wedge (j = t)];$$

we have just encoded all equations in (1) and (4). Next we encode the equations described in (2) and (3): define the relation  $\rho(\mathbf{H})$  by the formula

$$\phi_\rho \equiv \phi_\rho^{(2)} \vee \phi_\rho^{(3)}$$

where

$$\begin{aligned} \phi_\rho^{(2)}(i, j, i', j', i'', j'') &\equiv \{(i + 1 \leq j) \wedge \neg(i + 1 = j)\} \\ &\quad \bigwedge \{[(i + 1, j) \in \theta] \wedge (i' = i + 1) \wedge (j' = j + 1) \wedge (i'' = i') \wedge (j'' = j)\} \\ &\quad \bigvee \{[\neg((i + 1, j) \in \theta) \wedge (i' = t) \wedge (j' = t) \wedge (i'' = i + 1) \wedge (j'' = j)]\} \end{aligned}$$

and

$$\phi_\rho^{(3)}(i, j, i', j', i'', j'') \equiv [(i + 1 = j) \wedge (i' = t) \wedge (j' = t) \wedge (i'' = j) \wedge (j'' = j)].$$

[Notice the convenient use of the equation  $x_{t,t} = 0$  to encode equalities.]  $\square$

Similar arguments lead to an analogous result in which hardness is replaced by non-expressibility for fragments of Datalog.

**Theorem 4.2.** *Let  $\Gamma$  be a finite set of relations on  $A$  such that  $\mathbf{\Gamma}$  is a core, and let  $\mathbb{A} = \mathbb{A}(\Gamma)$ . Then:*

1. If  $\mathcal{V}(\mathbb{A})$  admits the unary or affine types, then  $\neg\text{CSP}(\Gamma)$  is not in Datalog.
2. If  $\mathcal{V}(\mathbb{A})$  omits the unary type, but admits the semilattice type, then  $\neg\text{CSP}(\Gamma)$  is not in linear Datalog.
3. If  $\mathcal{V}(\mathbb{A})$  omits the unary and semilattice types, but admits the lattice type, then  $\neg\text{CSP}(\Gamma)$  is not in symmetric Datalog.

Note that (1) is shown in [24]. As in the previous proof, let  $\theta = \{(x, y, z) : (y \wedge z) \rightarrow x\}$ : statement (2) relies on the fact that  $\neg\text{CSP}(\{\theta, \{0\}, \{1\}\})$  is not definable in linear Datalog<sup>6</sup> [1]. Statement (3) follows from the fact that  $\neg\text{CSP}(\{\leq, \{0\}, \{1\}\})$  is not definable in symmetric Datalog [14].

## 5. Applications

Because the algebraic criteria used in Theorems 4.1 and 4.2 are all decidable by results of Chapter 9 in [18] (see also [31]), they are a very convenient first step when studying the complexity of  $\text{CSP}(\Gamma)$  for a specific  $\Gamma$  or a specific class of them. We first show that our criteria match Allender et al.'s [2] description of the complexity of Boolean CSP's and line up exactly with the expressibility in restrictions of Datalog. We finally use them to study CSP linked to preprimal algebras.

### 5.1. Boolean CSP's

**Theorem 5.1.** *Let  $\Gamma$  be a set of relations on  $\{0, 1\}$  such that  $\Gamma$  is a core. Let  $\mathcal{V}$  denote the variety generated by  $\mathbb{A}(\Gamma)$ .*

1. *If  $\mathcal{V}$  admits the unary type then  $\text{CSP}(\Gamma)$  is NP-complete, and  $\neg\text{CSP}(\Gamma)$  is not expressible in Datalog.*
2. *If  $\mathcal{V}$  omits the unary type but admits the affine type, then  $\text{CSP}(\Gamma)$  is  $\oplus L$ -complete and  $\neg\text{CSP}(\Gamma)$  is not expressible in Datalog.*
3. *if  $\mathcal{V}$  admits only the Boolean type,  $\text{CSP}(\Gamma)$  is either first-order definable or L-complete; if  $\Gamma$  is finite,  $\neg\text{CSP}(\Gamma)$  is expressible in symmetric Datalog;*
4. *if  $\mathcal{V}$  omits the unary, affine and semilattice types, but admits the lattice type, then  $\text{CSP}(\Gamma)$  is NL-complete; if  $\Gamma$  is finite, then  $\neg\text{CSP}(\Gamma)$  is expressible in linear Datalog, but not in symmetric Datalog;*
5. *if  $\mathcal{V}$  omits the unary and affine types, but admits the semilattice type, then  $\text{CSP}(\Gamma)$  is P-complete; if  $\Gamma$  is finite then  $\neg\text{CSP}(\Gamma)$  is expressible in Datalog, but not in linear Datalog.*

**Proof.** All statements of non-expressibility follow directly from Theorem 4.2. We give the details of the correspondence between typesets of varieties and Boolean clones that will allow us to invoke the completeness results from [2]: all references to special operations on  $\{0, 1\}$  and clones in Post's lattice use the notation from that reference.

Fix  $\Gamma$  a (not necessarily finite) set of relations on  $\{0, 1\}$  such that  $\Gamma$  is a core, i.e. the clone  $\text{Pol}(\Gamma)$  contains no constant operation. Let  $\mathbb{A}$  denote the 2-element algebra associated to the problem  $\text{CSP}(\Gamma)$ , and let  $\mathcal{V}$  denote the variety generated by  $\mathbb{A}$ . We consider the possibilities for the clone of terms  $\text{Pol}(\Gamma)$  of the algebra  $\mathbb{A}$ .

We shall first consider the case where  $\mathbb{A}$  is not idempotent. Direct inspection of the Post lattice shows that there are exactly three clones that contain no constants but contain a non-trivial permutation: these are  $D$  (self-dual operations, i.e. those that commute with negation),  $L_3$  (the clone of affine operations) and  $N_2$  (the clone generated by the negation). If the clone of  $\mathbb{A}$  is  $N_2$  then  $\mathcal{V}$  admits the unary type; if the clone of  $\mathbb{A}$  is  $L_3$  then  $\mathcal{V}$  admits the affine type but not the unary type; if the clone of  $\mathbb{A}$  is  $D$  then  $\mathcal{V}$  admits only the Boolean type. In all 3 cases, the complexity of the associated problem is as expected: NP-complete for the unary type,  $\oplus L$ -complete for the affine type, and L-complete for the Boolean type. The clone  $D$  corresponds to the 2-colouring problem, whose complement is in symmetric Datalog, see [13].

Now suppose  $\mathbb{A}$  is idempotent. Clearly  $\mathbb{A}$  is strictly simple and hence we may use Lemma 1.3 and Corollary 1.2 to get a precise description of  $\mathbb{A}$ .

- Suppose that  $\mathcal{V}$  admits the unary type. This means that  $\mathbb{A}$  is a set (i.e. has no non-trivial operations) and hence its clone of terms is the clone of projections  $I_2$ . We conclude that  $\text{CSP}(\Gamma)$  is NP-complete.

- Suppose that  $\mathcal{V}$  omits the unary type but admits the affine type, so that  $\mathbb{A}$  is affine. Its clone of terms must then contain the operation  $x + y + z$  and all its operations are linear; thus the clone must be  $L_2$ . Hence  $\text{CSP}(\Gamma)$  is  $\oplus L$  complete.

- Suppose that  $\mathcal{V}$  omits both the unary and affine types but admits the semilattice type. This means that  $\mathbb{A}$  is term equivalent to a semilattice, and thus its clone is either  $V_2$  or  $E_2$ . Hence  $\text{CSP}(\Gamma)$  is P complete. If  $\Gamma$  is finite, the problem is HORN  $k$ -SAT which is in  $(1, k)$ -Datalog [15].

- Suppose that  $\mathcal{V}$  omits the unary, affine and semilattice types but admits the lattice type. Then  $\mathbb{A}$  is polynomially equivalent to the 2 element lattice. In particular, its clone  $C$  is idempotent and contains only monotone operations, i.e. is contained in  $M \cap R$ : this leaves only the clones  $S_{00}, S_{00}^n, D_2, S_{10}, S_{10}^n, M_2$ . It is a simple exercise to verify that adding the constants to any of the clones  $D_2, S_{00}$  and  $S_{10}$  will generate the clone of all monotone operations, i.e. each of the clones in the list is polynomially equivalent to the 2 element lattice. For each of these clones, the problem  $\text{CSP}(\Gamma)$  is NL-complete. For expressibility in linear Datalog: it suffices to do it for the clones  $S_{10}$  ( $S_{00}$  is identical) and  $D_2$ ; this last case follows from [11] since it contains a majority operation. As for  $S_{10}$ , the argument is as follows: (1) if we take  $\Gamma$  finite such that  $\text{Pol}(\Gamma)$  contains  $S_{10}$  then  $\text{Pol}(\Gamma)$  is above some  $S_{10}^n$  and thus contains the near-unanimity operation  $h_n$ ; then expressibility in linear Datalog follows from [11].

- Suppose that  $\mathcal{V}$  omits all types but the Boolean type. It follows from [29] Theorem 6.1 that there are three possibilities:

1.  $\mathbb{A}$  is a discriminator algebra, i.e. it admits the term operation  $t(x, y, z) = z$  if  $x = y$  and  $t(x, y, z) = x$  otherwise; this operation generates the clone  $D_1$ , hence the clone of terms of our algebra must be  $D_1$  or  $R$ . The problem  $\text{CSP}(\Gamma)$  is thus L complete or first-order definable.

<sup>6</sup> Interestingly, an equivalent statement was obtained thirteen years earlier by Cook and Sethi [10], although it is stated in terms of "path machines" whose relation to linear Datalog is not apparent at first glance.

2. There exists an element, let's suppose without loss of generality that it is 0, and an integer  $k$  such that the operations in the clone are precisely those idempotent operations that preserve the relation

$$R_k = \{(a_1, \dots, a_k) : \exists i a_i = 0\}.$$

We claim that the clones in question are the  $S_{12}^k$  and that in fact for each  $k$  we have  $S_{12}^k = \text{Pol}(R_k)$ . Indeed it is trivial to verify that the operation  $x \wedge (y \vee \bar{z})$  preserves each  $R_k$  and hence

$$S_{12} \subseteq \text{Pol}(R_k) \subseteq R$$

so obviously  $\text{Pol}(R_k)$  is one of the  $S_{12}^n$ . The rest is easy: the operation  $h_k$  is a  $k + 1$ -ary near-unanimity operation which clearly cannot preserve  $R_{k+1}$  since this relation is not determined by its projections onto  $k$  factors; hence  $S_{12}^k \neq \text{Pol}(R_{k+1})$ . On the other hand, a simple application of the pigeonhole principle shows that  $h_k$  preserves  $R_k$ . It follows that the problem  $\text{CSP}(\Gamma)$  is either L-complete or first-order definable.

3. the clone is that of all idempotent operations preserving *all* the relations  $R_k$  described above, i.e. it is  $S_{12}$  (or  $S_{02}$ ). It follows that the problem  $\text{CSP}(\Gamma)$  is L-complete.

If  $\Gamma$  is finite, then we are in case (1) or (2) and expressibility in symmetric Datalog follows from [13].  $\square$

### 5.2. Preprimal algebras

We now use our results to investigate the descriptive and computational complexity of CSP's whose associated algebra is the full idempotent reduct of a preprimal algebra. A finite algebra  $\mathbb{A}$  is *preprimal* if its clone of term operations is maximal in the lattice of clones, i.e. is properly contained in the set of all operations on the base set  $A$  but there is no clone strictly between these. Maximal clones satisfy remarkable properties, for instance every clone is contained in some maximal clone and they are finite in number. They were completely classified by I.G. Rosenberg (see [12]), thereby furnishing an explicit criterion to determine if a set of operations generates all operations on a finite set  $A$  by composition. Alternatively, one may view CSP's whose algebra is preprimal as those whose constraint language is minimal, in the sense that it is non-trivial but every one of its constraints can be inferred from any other non-trivial constraint in the language. It is easy to see that any maximal clone may be expressed in the form  $\text{Pol}(\theta)$  for some relation  $\theta$ ; we shall investigate problems  $\text{CSP}(\Gamma)$  where  $\text{Pol}(\Gamma) = \text{Pol}(\{\theta\} \cup \{a\} : a \in A)$ , i.e. such that the associated algebra of the problem  $\text{CSP}(\Gamma)$  is the full idempotent reduct of a preprimal algebra. We follow Rosenberg's classification of the relations  $\theta$  that yield maximal clones, see pages 230-231 of [12]. We also require an effective characterisation of FO definable CSP's from [22]. Let  $\mathbf{G}$  be a relational structure and let  $a, b \in G$ . We say that  $b$  *dominates*  $a$  in  $\mathbf{G}$  if for any basic relation  $R$  of  $\mathbf{G}$ , and any tuple  $t \in R$ , replacement of any occurrence of  $a$  by  $b$  in  $t$  will yield a tuple of  $R$ . If  $\mathbf{\Gamma}$  is a relational structure on  $A$ , we say that the structure  $\mathbf{\Gamma}^2$  *dismantles to the diagonal* if one may obtain, by successive removals of dominated elements of  $\mathbf{\Gamma}^2$ , the diagonal  $\{(a, a) : a \in A\}$ .

**Lemma 5.2** ([22]). *Let  $\Gamma$  be a set of relations such that  $\mathbf{\Gamma}$  is a core. Then  $\text{CSP}(\Gamma)$  is first-order expressible if and only if  $\mathbf{\Gamma}^2$  dismantles to the diagonal.*

Rosenberg's classification identifies six cases, listed below, for relations  $\theta$  that yield maximal clones.

(P) (Permutation) Here  $\theta = \pi^\circ = \{(a, \pi(a)) : a \in A\}$  for some permutation  $\pi$  which is fixed point free and of prime order. In this case  $\neg\text{CSP}(\Gamma)$  is expressible in symmetric Datalog by [13]. In particular  $\text{CSP}(\Gamma)$  is in L and in fact is L-complete: it is easy to show that  $\mathbf{\Gamma}^2$  does not dismantle to the diagonal and thus that  $\text{CSP}(\Gamma)$  is not FO-definable and is L-hard by Theorem 3.2.

(E) (Equivalence) Here  $\theta$  is a non-trivial equivalence relation on  $A$ ; following [13] the problem  $\neg\text{CSP}(\Gamma)$  is expressible in symmetric Datalog. Hence  $\text{CSP}(\Gamma)$  is in L, and again L-complete because one can also show that  $\mathbf{\Gamma}^2$  does not dismantle to the diagonal.

(A) (Affine) In this case  $\theta = \{(a, b, c, d) : a + b = c + d\}$  where  $\langle A; +, 0 \rangle$  is some Abelian  $p$ -group for some prime  $p$ . Notice that the associated algebra is affine (in the sense defined earlier) and so the variety it generates admits the affine type, and hence by Theorems 4.2 and 4.1  $\text{CSP}(\Gamma)$  is not in Datalog, and it is  $\text{Mod}_p$ -L-hard and in fact  $\text{Mod}_p$ -L-complete.

(C) (Central) Here  $\theta$  is a  $k$ -ary relation ( $k \geq 1$ ) different from  $A^k$  that must (among other things) have a *central* element, i.e. there is some  $c \in A$  such that  $\theta$  contains every tuple with an occurrence of  $c$ . In that case  $\mathbf{\Gamma}^2$  does dismantle to the diagonal and  $\text{CSP}(\Gamma)$  is FO-definable. It follows from Theorem 5 of [13] that  $\neg\text{CSP}(\Gamma)$  is in symmetric Datalog, and if  $\Gamma$  does not contain a so-called *biredundant relation* then  $\text{CSP}(\Gamma)$  is actually first-order definable [22].

(R) (Regular) Here  $\theta$  is a  $k$ -ary ( $k \geq 3$ ) *regular* relation defined as follows. Let  $\mathbf{S}$  denote the structure with universe  $\{1, \dots, k\}$  and one basic relation  $\theta(\mathbf{S})$  of arity  $k$ , consisting of all tuples  $(x_1, \dots, x_k)$  with at least one repeated coordinate. Operations that preserve this relation are known to be the following: all non-surjective operations and all essentially unary operations i.e. that depend on only one variable [12]. In particular, no non-trivial idempotent operation preserves  $\theta(\mathbf{S})$ . For any positive integer  $m$  let  $\mathbf{S}^m$  denote the  $m$ -th power of this structure. A  $k$ -ary relation on the set  $A$  is *regular* if there exists some positive integer  $m$ , and a surjective map  $\mu$  from  $A$  to  $\mathbf{S}^m$  such that  $\theta = \mu^{-1}(\theta(\mathbf{S}^m))$ . Clearly, in this case, the structure  $\langle A; \theta \rangle$  retracts onto  $\mathbf{S}^m$ , and it is easy to see that  $\mathbf{S}^m$  retracts onto  $\mathbf{S}$  ( $\mathbf{S}$  embeds in  $\mathbf{S}^m$  via the map  $x \mapsto (x, 1, \dots, 1)$ ). From results in [24], the relation  $\theta$  cannot be invariant under a so-called Taylor operation, and thus  $\mathbb{A}(\Gamma)$  generates a variety that admits the unary type and  $\text{CSP}(\Gamma)$  is NP-complete.

(O) (Order) In this last case,  $\theta$  is a bounded order relation, i.e. a reflexive, antisymmetric, transitive relation  $\leq$  with elements 0 and 1 such that  $0 \leq x \leq 1$  for all  $x \in A$ . In that case the variety generated by the associated algebra admits type 4 or 5 and hence by Theorems 4.1 and 4.2  $\neg\text{CSP}(\Gamma)$  is not expressible in symmetric Datalog, and  $\text{CSP}(\Gamma)$  is NL-hard. There is not much that is known at this time on these CSP's, either from the algebraic or the complexity point of view: the class of so-called order-primal algebras is vast and quite complex. There are posets for which the problem is NP-complete, others for which the problem is NL-complete: the examples known to be in NL have their complement definable in linear Datalog. It is also possible to construct, from a non-bounded example found in [24], a bounded poset whose associated problem is tractable and  $\text{Mod}_p\text{L}$ -hard for some  $p$ . Similarly, from an example found in [23], one may construct a tractable example whose variety admits type 5 and hence is P-complete.

Our case analysis can be summarized as follows.

**Theorem 5.3.** *Let  $\Gamma$  be a finite set of relations such that  $\text{Pol}(\Gamma) = \text{Pol}(\{\theta\} \cup \{\{a\} : a \in A\})$  where  $\text{Pol}(\theta)$  is a maximal clone. If the maximal clone is of type (P), (E), (A), (C), (R), (O), then  $\text{CSP}(\Gamma)$  satisfies the properties given in the following:*

**(P)** (Permutation) Symmetric Datalog; L-complete.

**(E)** (Equivalence) Symmetric Datalog; L-complete.

**(A)** (Affine) not Datalog;  $\text{Mod}_p\text{L}$ -complete for some prime  $p$ .

**(C)** (Central) Symmetric Datalog; first-order definable if  $\Gamma$  contains no biredundant relation, L-complete otherwise.

**(R)** (Regular) NP-complete.

**(O)** (Order) not in symmetric Datalog; NL-hard; some cases are known to be NP-complete, some known to be NL-complete, some P-complete.

## 6. Conclusion

Our results demonstrate that the algebraic approach to CSP is sound when studying the expressive power of meaningful fragments of Datalog (see also [4]) and can prove quite powerful to answer questions about the complexity of CSP beyond the P vs. NP dichotomy. It is certainly tempting in light of the Boolean case (Theorem 5.1) to conjecture that some form of converse of Theorems 4.1 and 4.2 holds. It was already conjectured in [7] that  $\text{CSP}(\Gamma)$  is tractable if  $\mathcal{V}(\mathbb{A}(\Gamma))$  omits the unary type and this was verified for domains of size three and for list-CSP problems [5,6]. These two special cases seem to be the most promising testing ground for exploring a refined analysis of the complexity of CSP.

## Acknowledgements

We thank Albert Atserias, Victor Dalmau, László Egri, Andrei Krokhin, Pierre McKenzie, Matt Valeriote, Heribert Vollmer for helpful discussions. We are also grateful to the organizers of the Dagstuhl Seminar on the Complexity of Constraints where much of this work was completed. Finally we thank the anonymous referees for their judicious comments.

## References

- [1] F.N. Afrati, S.S. Cosmadakis, Expressiveness of restricted recursive queries, in: Proc. 21st ACM Symp. on Theory of Computing, STOC'89, 1989, pp. 113–126.
- [2] E. Allender, M. Bauland, N. Immerman, H. Schnoor, H. Vollmer, The complexity of satisfiability problems: Refining Schaefer's theorem, in: Proc. 30th Math. Found. of Comp. Sci., MFCS'05, 2005, pp. 71–82.
- [3] A. Atserias, On digraph coloring problems and treewidth duality, in: Proc. 21st Conf. on Logic in Comp. Sci., LICS'05, 2005, pp. 106–115.
- [4] A. Atserias, A. Bulatov, A. Dawar, Affine systems of equations and counting infinitary logic, in: Proc. 34th Int. Conf. on Automata, Languages and Programming, ICALP'07, 2007, pp. 558–570.
- [5] A. Bulatov, A dichotomy theorem for constraints on a three-element set, in: Proc. of 43rd Foundations of Comp. Sci., FOCS'02, 2002, pp. 649–658.
- [6] A. Bulatov, Tractable conservative constraint satisfaction problems, in: 18th IEEE Symp. on Logic in Comp. Sci., LICS 2003, 2003, pp. 321–331.
- [7] A. Bulatov, A. Krokhin, P. Jeavons, Constraint satisfaction problems and finite algebras, in: Proc. 27th Int. Coll. on Automata, Languages and Programming, ICALP'00, 2000, pp. 272–282.
- [8] G. Buntrock, C. Damm, U. Hertrampf, C. Meinel, Structure and importance of logspace-MOD classes, Math. Syst. Theory 25 (3) (1992) 223–237.
- [9] D. Cohen, P.G. Jeavons, The complexity of constraint languages, in: Handbook of Constraint Programming, 2006.
- [10] S.A. Cook, R. Sethi, Storage requirements for deterministic polynomial time recognizable languages, J. Comput. System Sci. 13 (1) (1976) 25–37.
- [11] V. Dalmau, Linear Datalog and bounded path duality of relational structures, Logical Methods Comput. Sci. 1 (1) (2005).
- [12] K. Denecke, S.L. Wismath, Universal Algebra and Applications in Theoretical Computer Science, CRC/C&H, 2002.
- [13] L. Egri, B. Larose, P. Tesson, Symmetric datalog and constraint satisfaction problems in logspace, in: Proc. 22nd Symp. on Logic in Comp. Sci., LICS-07, pp. 193–202.
- [14] L. Egri, B. Larose, P. Tesson, Symmetric datalog and constraint satisfaction problems in logspace, in: Proc. 35th Int. Conf. on Automata, Languages and Programming, ICALP'08, 2008.
- [15] T. Feder, M.Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory, SIAM J. Comput. 28 (1) (1999) 57–104.
- [16] T. Feder, M.Y. Vardi, Homomorphism closed vs. existential positive, in: Proc. 18th Symp. on Logic in Comp. Sci., LICS 2003, 2003, pp. 311–320.
- [17] M. Furst, J.B. Saxe, M. Sipser, Parity, circuits, and the polynomial-time hierarchy, Math. Syst. Theory 17 (1) (1984) 13–27.
- [18] D. Hobby, R. McKenzie, The Structure of Finite Algebras, in: Contemporary Mathematics, vol. 76, American Mathematical Society, 1988.
- [19] N. Immerman, Descriptive Complexity, in: Graduate Texts in Computer Science, Springer, 1999.
- [20] P. Jeavons, On the algebraic structure of combinatorial problems, Theoret. Comput. Sci. 200 (1–2) (1998) 185–204.
- [21] P.G. Kolaitis, M.Y. Vardi, On the expressive power of datalog: Tools and a case study, J. Comput. System Sci. 51 (1) (1995) 110–134.
- [22] B. Larose, C. Loten, C. Tardif, A characterization of first-order constraint satisfaction problems, Logical Methods Comput. Sci. 3 (4) (2007).
- [23] B. Larose, L. Zádori, Finite posets and topological spaces in locally finite varieties, Algebra Universalis 52 (2–3) (2005) 119–136.
- [24] B. Larose, L. Zádori, Bounded width problems and algebras, Algebra Universalis 56 (3–4) (2007) 439–466.

- [25] L. Libkin, *Elements of Finite Model Theory*, Springer Verlag, 2004.
- [26] O. Reingold, Undirected st-connectivity in log-space, in: Proc. 37th ACM Symp. on Theory of Computing, STOC'05, 2005, pp. 376–385.
- [27] B. Rossman, Existential positive types and preservation under homomorphisms, in: Proc. 21st Conf. on Logic in Comp. Sci., LICS'05, 2005, pp. 467–476.
- [28] T.J. Schaefer, The complexity of satisfiability problems, in: Proc. 10th ACM STOC, 1978, pp. 216–226.
- [29] A. Szendrei, A survey on strictly simple algebras and minimal varieties, in: *Research and Exposition in Mathematics*, Heldermann Verlag, 1992, pp. 209–239.
- [30] J. Ullman, *Database and Knowledge-Base Systems*, Computer Science Press, 1989.
- [31] M. Valeriote, A subalgebra intersection property for congruence distributive varieties, *Canad. J. Math.*, 2007, in press (<http://journals.cms.math.ca/cgi-bin/vault/viewprepub/valeriote7024.prepub>).