Electronic Colloquium on Computational Complexity, Report No. 25 (2007)

Universal Algebra and Hardness Results for Constraint Satisfaction Problems ⋆

Benoît Larose[1] and Pascal Tesson[2]

[1] Department of Mathematics and Statistics, Concordia University
larose@mathstat.concordia.ca
[2] Département d'Informatique et de Génie Logiciel, Université Laval
pascal.tesson@ift.ulaval.ca

Abstract. We present algebraic conditions on constraint languages Γ that ensure the hardness of the constraint satisfaction problem CSP(Γ) for complexity classes L, NL, P, NP and Mod$_p$L. These criteria also give non-expressibility results for various restrictions of Datalog. Furthermore, we show that if CSP(Γ) 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 CSPs. The results pave the way for a refinement of the dichotomy conjecture stating that each CSP(Γ) lies in P or is NP-complete and they match the recent classification of [1] for Boolean CSP. We also infer a partial classification theorem for the complexity of CSP(Γ) when the associated algebra of Γ is the idempotent reduct of a preprimal algebra.

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* Γ, i.e. some set of finitary relations over a fixed domain. In an instance of CSP(Γ), all constraints are of the form $(x_{i_1}, \ldots, x_{i_k}) \in R_j$ for some $R_j \in \Gamma$.

In their seminal work [9], Feder and Vardi conjectured that each CSP(Γ) 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 [18] on the complexity of Generalized Satisfiability which states that CSP(Γ) is either in P or is NP-complete for any constraint language Γ 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.

⋆ Research supported by NSERC, FQRNT and CRM. We thank Albert Atserias, Victor Dalmau, Laszlo Egri, Andrei Krokhin, Matt Valeriote and Heribert Vollmer for helpful discussions.

ISSN 1433-8092

One such angle of attack relies on universal algebra: there is a natural way to associate to a set of relations Γ an algebra $\mathbb{A}(\Gamma)$ whose operations are the functions that *preserve* the relations in Γ and one can show that the complexity of CSP(Γ) 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 and the identification of wide classes of tractable CSP (see [4]).

A different, descriptive complexity approach has consisted in classifying CSPs according to the sophistication of the logical apparatus required to define the set of negative instances of CSP(Γ). It was noted early on that when $\neg$CSP(Γ) is definable in the database-inspired logic programming language Datalog then CSP(Γ) lies in P [9] 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 and linear Datalog and CSPs solvable in, respectively, logarithmic space (L) and non-deterministic logspace (NL) [5, 8]. From a logical perspective, the simplest class of CSPs are those which are first-order definable and recent work has provided a precise characterization for them [2, 15].

From a complexity-theoretic perspective, the classification of CSP(Γ) as "tractable" is rather coarse. Ultimately, one would expect that every CSP(Γ) 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 in fact be refined: each CSP(Γ) over the Boolean domain is either **FO**-definable or is complete under AC^0 isomorphisms for one of the classes L, NL, $\oplus$L, P or NP [1].

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 CSPs. As we recall in Section 2, the algebraic angle of attack relies on a number of basic reductions from CSP(Λ) to CSP(Γ) when Γ and Λ are constraint languages when the algebra $\mathbb{A}(\Lambda)$ lies in the variety generated by $\mathbb{A}(\Gamma)$ [3, 4]. When the sole objective is to classify CSP(Γ) 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 them 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 tracatable CSPs. In Section 3, we begin by proving that if CSP(Γ) is not first-order definable, then the problem is in fact L-hard. 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 Section 4, we use tame congruence theory and deep classification results of idempotent strictly simple algebras to obtain a number of hardness results for CSP(Γ). Specifically, we consider the variety $\mathcal{V}$ generated by $\mathbb{A}(\Gamma)$ and give sufficient conditions on $\mathcal{V}$ for CSP(Γ) to be NL-hard, Mod$_p$L-hard and P-hard. These also translate into necessary conditions for $\neg$CSP(Γ) to be definable in Datalog, linear Datalog and symmetric Datalog. For a given $\mathit{\Gamma}$ it is possible to decide whether or not $\mathbf{V}$ fits each of these criteria [20]. In Section 5, we demonstrate the usefulness of the results by revisiting the results of [1] on CSPs over the boolean domain and by classifying the complexity of a number of CSP($\mathit{\Gamma}$) when $\mathbb{A}(\mathit{\Gamma})$ is the idempotent reduct of a preprimal algebra.

Because of space constraints, technical proofs appear in the appendix.

1 Preliminaries

We first set the notation and present the required basics. We refer the reader to [11] and [7] for algebraic and clone-theoretic results, to [12] for an introduction to finite model theory and descriptive complexity and to [4] for a survey on the algebraic approach to CSP.

Let $\sigma = \{R_1, \ldots, R_r, c_1, \ldots, c_s\}$ 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 σ is a tuple $\mathbf{H} = \langle H; R_1(\mathbf{H}), \ldots, R_r(\mathbf{H}), c_1(\mathbf{H}), \ldots, c_s(\mathbf{H})\rangle$ 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 use the usual convention of using $G, H, \ldots$ to denote the universe of the structure $\mathbf{G}, \mathbf{H}, \ldots$. Unless otherwise mentioned the signatures we deal with in this paper are purely relational (i.e. without constant symbols).

Let σ be a (relational) signature. Given two σ-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

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

Two structures $\mathbf{H}$ and $\mathbf{H}'$ are *homomorphically equivalent* if there exist homomorphisms $\mathbf{H} \to \mathbf{H}'$ and $\mathbf{H}' \to \mathbf{H}$. A structure $\mathbf{H}$ is a *core* if the only homomorphisms $\mathbf{H} \to \mathbf{H}$ 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 $\mathit{\Gamma}$ 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 $\mathit{\Gamma}$ and let σ be its signature. We denote as Hom($\mathbf{\Gamma}$) the class of all finite structures of type σ that admit a homomorphism to $\mathbf{\Gamma}$. In this setting $\mathbf{\Gamma}$ is called the *target structure.* Alternatively, we may use the notation CSP($\mathit{\Gamma}$) for this decision problem: indeed the constraints in an instance of CSP($\mathit{\Gamma}$) can be regarded as defining a σ-structure $\mathbf{C}$ on the set of variables and a satisfying assignment is a homomorphism from $\mathbf{C}$ to $\mathbf{\Gamma}$. Obviously, if $\mathbf{\Gamma}'$ is the core of $\mathbf{\Gamma}$ then Hom($\mathbf{\Gamma}$) = Hom($\mathbf{\Gamma}'$). 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 θ on A (equivalently, we say that θ is *invariant* *under f*) if the following holds: given any matrix M of size $k \times n$ whose columns are in θ, applying f to the rows of M will produce a k-tuple in θ. Given a set Γ of relations on A, $Pol(\Gamma)$ denotes the set of all operations on A that preserve all relations in Γ.

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 Γ over A, we denote by $\mathbb{A}(\Gamma)$ the algebra $\langle A; Pol(\Gamma)\rangle$, and call it the *algebra associated to* $CSP(\Gamma)$.

The terms (polynomials) of an algebra are the operations that can be built from its basic operation (and the constants) using composition and projections. Two algebras are term (polynomially) equivalent if they have the same terms (polynomials). An operation f is *idempotent* if it satisfies $f(x, \ldots, x) = x$ for all x. The *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 [11]. A class of similar algebras 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 $\mathcal{V}(\mathbb{A}) = HSP(\mathbb{A})$, 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 [11], is a powerful tool for the analysis of finite algebras. Every finite algebra has a *typeset*, which describes the local behaviour of the algebra, which 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 (omitted) by a variety, i.e. those types that (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 and has no non-trivial subalgebras (a subalgebra is trivial if it is either the algebra itself or is 1-element.) Because a strictly simple algebra is simple it has a unique type from 1 to 5 associated to it. The next lemma (Lemma 3.1 [20]) will allow us to connect typesets of varieties to the complexity of CSP's:

Lemma 1.1. *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 $HS(\mathbb{A})$ where "at most i" refers to the ordering* $1 < 2 < 3 > 4 > 5 > 1$.

Szendrei has characterised all idempotent strictly simple algebras, ([19] 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 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 idempotent reduct of a module.

Lemma 1.2. *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 it is of semilattice (type 5) 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 1.1. *Let $\mathbb{A}$ be a finite, idempotent, strictly simple algebra.*

1. *If $\mathbb{A}$ is of 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 $\mathbb{A}$ is of semilattice type, then up to isomorphism the universe of $\mathbb{A}$ is $\{0,1\}$ and the relation $\theta = \{(x,y,z) : (y \wedge z) \to x\}$ is a subalgebra of $\mathbb{A}^3$;*
3. *if $\mathbb{A}$ is of 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 $\mathbb{A}^2$.*

1.2 Fragments of Datalog

Datalog was originally introduced as a database query language. We view it here simply as a means to define sets of σ-structures. A Datalog program over the signature σ consists of a finite set of *rules* of the form $h \leftarrow b_1 \wedge \ldots \wedge b_k$ where each of the b_i and h are atomic formulas of the form $R(x_{j_1}, \ldots, x_{j_r})$. We distinguish two types of relational predicates occurring in the program: predicates R that occur at least once in the head of a rule are called *intensional database predicates* (IDBs) and are not part of σ. The other predicates which occur only in the body of a rule are called *extensional database predicates* and must all lie in σ. Precise definitions of the semantics of Datalog can be found in [14, 5, 8]: we simply illustrate the basics of the formalism with the following example.

Let σ be a signature consisting of a single binary relational symbol E (so that a σ-structure is a graph) and consider the Datalog program consisting of the rules (1) $P(x,y) \leftarrow E(x,y)$ (2) $P(x,y) \leftarrow P(x,z) \wedge P(z,y)$ and (3) $G(x) \leftarrow P(x,x)$. The program 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 and we say that a σ-structure is accepted by the program if that target IDB is non-empty. The program above 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 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 such that each recursive rule $\mathcal{R}$ is accompanied by its symmetric $\mathcal{R}^r$, where $\mathcal{R}^r$ is obtained by exchanging from $\mathcal{R}$ by exchanging the roles of the IDBs in its head and body. The symmetric of the above rule would be $P(x, z) \leftarrow P(x, y) \wedge E(z, y)$.

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 [2] shows that CSP(Γ) is definable by a first-order sentence iff $\neg$CSP(Γ) is definable by a non-recursive Datalog program and consequently all such CSP(Γ) are solvable in co-NLogtime. Moreover, expressibility of $\neg$CSP(Γ) in symmetric, linear and general Datalog is a sufficient (and wide-encompassing) condition for CSP(Γ) to lie in respectively L [8], NL [5] and P [9].

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 θ is *irredundant* if for each two distinct coordinates i and j there exists a tuple $\overline{x}$ of θ with $x_i \neq x_j$.

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

1. *Let $\mathbb{C}$ be a finite algebra in $\mathcal{V}(\mathbb{A})$, and let Γ_0 be a finite set of relations invariant under the basic operations of $\mathbb{C}$. Then there exists a logspace many-one reduction of* CSP(Γ_0) *to* CSP(Γ). *Furthermore, if $\neg CSP(\Gamma)$ is expressible in (linear, symmetric) Datalog, then so is $\neg CSP(\Gamma_0)$.*
2. *If furthermore $\mathbb{C} \in HS(\mathbb{A})$ and the relations in Γ_0 are irredundant, then the above reduction is first-order.*

The proof, although not conceptually difficult, is technical and is given in the appendix in full detail. The constructions are given for ten basic reductions which can be composed to obtain the two claims above: their principles are not new [13, 3] although most were never explicitly shown to be first order or to preserve expressibility in the linear and symmetric fragments of Datalog (the case of Datalog is treated in [17]). 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 CSP($\Gamma$) is first-order definable then CSP(Γ_0) is L-complete (see e.g. [8]) and so there can be no first-order reduction from CSP(Γ_0) to CSP(Γ).

3 CSP's that are not FO are L-hard

In this section we show that for every finite set Γ, if $CSP(\Gamma)$ is not first-order expressible then it is L-hard. We require a characterisation of first-order definable CSP's from [15]. Consider the signature $\sigma = \{R_1, \ldots, R_r\}$ where R_i is a relational symbol of arity a_i. For an integer n the *n-link* of type σ is the σ-structure

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

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

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

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

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

Let π_1 and π_2 denote the two projections from A^2 to A.

Lemma 3.1 ([15]). *Let Γ be a finite set of relation on A such that $\mathbf{\Gamma}$ is a core. Then $CSP(\Gamma)$ is first-order definable if and only if for some n there exists a homomorphism $P : \mathbf{L}_n \to \mathbf{\Gamma}^{(\mathbf{\Gamma}^2)}$ such that $P(0) = \pi_1$ and $P(n) = \pi_2$.*

Theorem 3.1. *For any finite constraint language Γ the problem $CSP(\Gamma)$ is either first-order definable or is L-hard.*

Proof (Sketch). We assume that $\mathbf{\Gamma}$ is a core. Let $\sigma = \{R_1, \ldots, R_r\}$ be the signature of the structure $\mathbf{\Gamma}$ and let $\Gamma' = \Gamma \cup \{\{a\} : a \in A\}$. For $a \in A$ let S_a be a relational symbol for $\{a\}$, and let $\sigma' = \{R_1, \ldots, R_r\} \cup \{S_a : a \in A\}$ be the signature of the structure $\mathbf{\Gamma'}$. It suffices to show that if $CSP(\Gamma)$ is not first-order definable then $CSP(\Gamma')$ is L-hard (see Lemma 6.4): for this we construct a first-order reduction of NOT *st*-connectivity (which is L-hard under first-order reductions [12]) to $CSP(\Gamma')$. 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 σ' into τ assigning to each graph $\mathbf{G}$ with distinguished vertices s and t a structure $\mathbf{K} = \mathcal{I}(\mathbf{G})$ of type σ' such that $\mathbf{K}$ admits a homomorphism to $\mathbf{\Gamma'}$ precisely when s and t are *not* connected in $\mathbf{G}$.

It is clear that the following defines a symmetric relation $\sim$ on $\mathbf{\Gamma}^{(\mathbf{\Gamma}^2)}$: let $g \sim h$ if there exists a homomorphism $F : \mathbf{L}_1 \to \mathbf{\Gamma}^{(\mathbf{\Gamma}^2)}$ 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 $\mathbf{\Gamma}$, $CSP(\Gamma)$ is first-order definable precisely when the projections are connected in the graph defined by $\sim$.

We simply describe the reduction and argue in the appendix that it is indeed first-order. Given a graph $\mathbf{G}$ with specified vertices s and t, we construct a σ-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 σ-structure $\mathbf{H}\times\mathbf{\Gamma}^2$, that we transform into the σ'-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 $CSP(\Gamma')$. Suppose there is a homomorphism f from $\mathbf{K}$ to $\mathbf{\Gamma}'$: in particular it is a homomorphism of σ-structures $f:\mathbf{H}\times\mathbf{\Gamma}^2\rightarrow\mathbf{\Gamma}$, which, by the natural property of products, induces a homomorphism F from $\mathbf{H}$ to $\mathbf{\Gamma}^{\mathbf{\Gamma}^2}$; 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 $CSP(\Gamma)$ is not first-order definable, there is no path in $\mathbf{\Gamma}$ between the projections, 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 $\mathbf{H}\times\mathbf{\Gamma}^2$ to $\mathbf{\Gamma}$ by setting $f(g,c,d)=c$ if there 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 $F:\mathbf{H}\rightarrow\mathbf{\Gamma}^{\mathbf{\Gamma}^2}$ induced by f maps all elements to one of two projections which are "loops" in any power structure.

4 Main Theorems

We present our two main theorems. The first provides sufficient algebraic criteria for the hardness of CSP(Γ) for a number of natural complexity classes. The second expresses these same lower bounds in descriptive complexity terms.

Theorem 4.1. *Let Γ 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 type, then* CSP(Γ) *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* CSP(Γ) *is Mod_pL -hard under FO reductions.*
3. *If* CSP(Γ) *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* CSP(Γ) *is NL-hard under FO reductions.*
5. *If $\mathcal{V}(\mathbb{A})$ omits the unary type, but admits the semilattice type, then* CSP(Γ) *is P-hard under FO reductions.*

Proof (Sketch). (3) is the content of Theorem 3.1, and (1) follows from a result of [3]: the reduction there is actually first-order by Theorem 2.1.

It follows from results in [11] that if $\mathbb{A}$ satisfies the hypothesis of one of (2), (4) or (5) then so does its idempotent reduct, which we denote by $\mathbb{B}$. By Lemma 1.1, if $\mathbb{B}$ satisfies the hypothesis (i) then there exists a strictly simple algebra $\mathbb{C}\in HS(\mathbb{A})$ of type (i). By Corollary 1.1 it means that, in case (2), there exists an Abelian group structure on the base set of $\mathbb{B}$ such that the 3-ary relation $\mu=\{(x,y,z):x+y=z\}$ is invariant under the operations of $\mathbb{B}$. Consider the set Γ' that consists of the relation μ, the unary relation $B=\{b\}$

where b is some non-zero element of $\mathbb{B}$ such that $pb = 0$ for some prime p, and the unary relation $Z = \{0\}$. Then by Theorem 2.1 there is a first-order reduction of CSP(Γ') to CSP(Γ) and by Lemma 10.1 (in the appendix) CSP(Γ) is mod-p L hard under FO reductions.

We proceed as in the other two cases. In case (4), we find an FO reduction of CSP(Γ') to CSP(Γ) where $\Gamma' = \{\leq, \{0\}, \{1\}\}$. There is a straightforward FO-reduction from the directed graph unreachability problem to CSP(Γ') and the former problem is NL-complete under first-order reductions [12].

In case (5) we find a first-order reduction of CSP(Γ') to CSP(Γ) where $\Gamma' = \{\theta, \{0\}, \{1\}\}$ where $\theta = \{(x, y, z) : (y \wedge z) \rightarrow x\}$. It is again straightforward to show that CSP(Γ') admits a natural FO reduction from Horn-3-Sat which is P-hard under FO-reductions [12].

Similar arguments lead to an analogous result in which hardness is replaced by non-expressibility for fragments of Datalog. Note that (1) is shown in [17], while (2) and (3) rely on the fact that $\neg$CSP($\{\leq, \{0\}, \{1\}\}$) and $\neg$CSP($\{\theta, \{0\}, \{1\}\}$) are not definable in, respectively, symmetric and linear Datalog [6].

Theorem 4.2. *Let Γ 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 $\neg CSP(\Gamma)$ is not in Datalog.*
2. *If $\mathcal{V}(\mathbb{A})$ omits the unary type, but admits the semilattice type, then $\neg CSP(\Gamma)$ is not in linear Datalog.*
3. *If $\mathcal{V}(\mathbb{A})$ omits the unary and semilattice types, but admits the lattice type, then $\neg CSP(\Gamma)$ is not in symmetric Datalog.*

5 Applications

Because the algebraic criteria used in Theorems 4.1 and 4.2 are all decidable [20], they are a very convenient first step when studying the complexity of CSP(Γ) for a specific Γ or a specific class of them. We first show that our criteria match Allender et al.'s [1] 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 CSPs linked to preprimal algebras.

5.1 Boolean CSP's

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

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

4. *if $\mathcal{V}$ omits the unary, affine and semilattice types, but admits the lattice type, then $CSP(\Gamma)$ is NL-complete; if Γ is finite, then $\neg CSP(\Gamma)$ 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 $CSP(\Gamma)$ is P-complete; if Γ* is finite *then $\neg CSP(\Gamma)$ is expressible in Datalog, but not in linear Datalog.*

5.2 Preprimal algebras

We now use our results to investigate the descriptive and computational complexity of CSP's whose associated algebra is the 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 [7]), 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 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 $Pol\,(\theta)$ for some relation θ; we shall investigate problems CSP(Γ) where $Pol(\Gamma) = Pol(\{\theta\} \cup \{\{a\} : a \in A\})$, i.e. such that the associated algebra of the problem $CSP(\Gamma)$ is the idempotent reduct of a preprimal algebra. We follow Rosenberg's classification of the relations θ that yield maximal clones, see pages 230-231 of [7]. We also require an effective characterisation of FO definable CSP's from [15]. 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 5.1 ([15]). *Let Γ be a set of relations such that $\mathbf{\Gamma}$ is a core. Then $CSP(\Gamma)$ is first-order expressible if and only if $\mathbf{\Gamma}^2$ dismantles to the diagonal.*

(P) (Permutation) Here $\theta = \pi^\circ$ for some permutation π which is fixed point free and of prime order. In this case $\neg CSP(\Gamma)$ is expressible in symmetric Datalog by [8]. In particular CSP(Γ) 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 CSP(Γ) is not FO-definable and L-hard by Theorem 3.1.

(E) (Equivalence) Here θ is a non-trivial equivalence relation on A; following [8] the problem $\neg CSP(\Gamma)$ is expressible in symmetric Datalog. Hence CSP(Γ) 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 CSP(Γ) is not in Datalog, and it is Mod$_p$L-hard and in fact Mod$_p$L-complete (see appendix).

(C) (Central) Here θ 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 θ contains every tuple with an occurrence of c. In that case $\mathbf{\Gamma}^2$ does dismantles to the diagonal and CSP(Γ) is FO-definable. It follows from Theorem 5 of [8] that $\neg CSP(\Gamma)$ is in symmetric Datalog, and if Γ does not contain a so-called *biredundant relation* then CSP(Γ) is actually first-order definable [15].

(R) (Regular) Here θ is a k-ary ($k \geq 3$) *regular* relation defined as follows. Let $\mathbf{S}$ denote the structure with universe $\{1, \ldots, k\}$ and one basic relation $\theta(\mathbf{S})$ of arity k, consisting of all tuples $(x_1, \ldots, x_k)$ 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 [7]. 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 μ 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 $x \mapsto (x, 1, \ldots, 1)$.) From results in [17], the relation θ cannot be invariant under a so-called Taylor operation, and thus $\mathbb{A}(\Gamma)$ generates a variety that admits the unary type and CSP(Γ) is NP-complete.

(O) (Order) In this last case, θ 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 4.2 and 4.1 $\neg CSP(\Gamma)$ is not expressible in symmetric Datalog, and CSP(Γ) 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 [17], a bounded poset whose associated problem is tractable and mod-p L hard. Similarly, from an example found in [16], 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 5.2. *Let Γ be a finite set of relations such that $Pol(\Gamma) = Pol(\{\theta\} \cup \{\{a\} : a \in A\})$ where $Pol(\theta)$ is a maximal clone. If the maximal clone is of type (P), (E), (A), (C), (R), (O), then $CSP(\Gamma)$ satisfies the properties given in the following:*

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

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

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

(C) *(Central) Symmetric Datalog; first-order definable if Γ 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.*

References

1. E. Allender, M. Bauland, N. Immerman, H. Schnoor, and H. Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. In *Proc. 30th Math. Found. of Comp. Sci. (MFCS'05)*, pages 71–82, 2005.
2. A. Atserias. On digraph coloring problems and treewidth duality. In *Proc. 21st Conf. on Logic in Comp. Sci. (LICS'05)*, pages 106–115, 2005.
3. A. Bulatov, A. Krokhin, and P. Jeavons. Constraint satisfaction problems and finite algebras. In *Proc. 27th Int. Coll. on Automata, Languages and Programming—ICALP'00*, pages 272–282, 2000.
4. D. Cohen and P. G. Jeavons. *The Complexity of Constraint Languages*, chapter 8. Elsevier, 2006.
5. V. Dalmau. Linear Datalog and bounded path duality of relational structures. *Logical Methods in Computer Science*, 1(1), 2005.
6. V. Dalmau, L. Egri, B. Larose, and P. Tesson. On the limits of expressivity of linear and symmetric Datalog. Document in preparation, 2007.
7. K. Denecke and S. L. Wismath. *Universal Algebra and Applications in Theoretical Computer Science*. CRC/C&H, 2002.
8. L. Egri, B. Larose, and P. Tesson. Symmetric datalog and constraint satisfaction problems in logspace. Submitted, 2007.
9. T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM J. on Computing*, 28(1):57–104, 1999.
10. T. Feder and M. Y. Vardi. Homomorphism closed vs. existential positive. In *Proc. 18th Symp. on Logic in Comp. Sci. (LICS 2003)*, pages 311–320, 2003.
11. D. Hobby and R. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, 1988.
12. N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1999.
13. P. Jeavons. On the algebraic structure of combinatorial problems. *Theor. Comput. Sci.*, 200(1-2):185–204, 1998.
14. P. G. Kolaitis and M. Y. Vardi. On the expressive power of datalog: Tools and a case study. *J. Comput. Syst. Sci.*, 51(1):110–134, 1995.
15. B. Larose, C. Loten, and C. Tardif. A characterization of first-order constraint satisfaction problems. In *Proc. 21st Symp. on Logic in Comp. Sci. (LICS-06)*, pages 201–210, 2006.
16. B. Larose and L. Zádori. Finite posets and topological spaces in locally finite varieties. *Algebra Universalis*, 52(2-3):119–136, 2005.
17. B. Larose and L. Zádori. Bounded width problems and algebras. *Algebra Universalis*, 2006.
18. T. J. Schaefer. The complexity of satisfiability problems. In *Proc. 10^{th} ACM STOC*, pages 216–226, 1978.
19. A. Szendrei. *A survey on strictly simple algebras and minimal varieties*, pages 209–239. Research and Exposition in Mathematics. Heldermann Verlag, 1992.
20. M. Valeriote. A subalgebra intersection property for congruence distributive varieties. *Canadian J. of Math.*, 2007.

Appendix

We provide in this appendix the technical arguments which were left out of the main body of the paper.

6 Reductions

The next two sections provide the intermediate results required to prove Theorem 2.1. We first introduce some terminology. We shall use the notion of *first-order interpretation with parameters* which Atserias used in a similar context in [2].

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

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

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

and whose interpretation for R_i is

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

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

Definition. Let σ and τ be finite relational vocabularies, let $\mathcal{C}$ be a class of σ-structures and let $\mathcal{D}$ be a class of τ-structures closed under isomorphisms. We say that a first-order (resp. quantifier-free) interpretation $\mathcal{I}$ with p parameters of τ in σ is a *first-order (resp. quantifier-free) reduction from $\mathcal{C}$ to $\mathcal{D}$* if for every σ-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. ϕ_U is quantifier-free;
2. for every θ in τ, ϕ_θ 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.

6.1 The algebraic reductions

Lemma 6.1. *Let Γ_0 be a finite set of relations on C, let $\mu : B \to C$ be a surjective map. Let $\Gamma_1 = \{\mu^{-1}(\theta) : \theta \in \Gamma_0\}$. Then* $\mathrm{Hom}(\mathbf{\Gamma}_0) = \mathrm{Hom}(\mathbf{\Gamma}_1)$, *in particular there is a positive quantifier-free reduction of* $\mathrm{CSP}(\Gamma_0)$ *to* $\mathrm{CSP}(\Gamma_1)$.

Proof. This is straightforward: let C' denote a subset of B that maps bijectively to C via μ; then the substructure of $\mathbf{\Gamma}_1$ induced by C' is obviously isomorphic to $\mathbf{\Gamma}_0$ via μ; hence $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_1$ are homomorphically equivalent hence $\mathrm{Hom}(\mathbf{\Gamma}_0) = \mathrm{Hom}(\mathbf{\Gamma}_1)$. The last statement is trivial.

Lemma 6.2. *Let Γ_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 Γ_1 are viewed as relations on A. Then there is a positive quantifier-free reduction of* $\mathrm{CSP}(\Gamma_1)$ *to* $\mathrm{CSP}(\Gamma_2)$.

Proof. This is also quite easy: let $\sigma = \{R_1, \ldots, R_s\}$ be the type of the structure $\mathbf{\Gamma}_1$ and let $\tau = \{R_0, R_1, \ldots, R_s\}$ be the type of the structure $\mathbf{\Gamma}_2$. Let $\mathbf{G}$ be a structure of type σ, and define the structure $\mathbf{G}'$ of type τ 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 $\mathbf{\Gamma}_1$ if and only if there is one from $\mathbf{G}'$ to $\mathbf{\Gamma}_2$; furthermore the reduction is clearly quantifier-free and positive.

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

$$\widetilde{\theta} = \{(x_1^1, x_1^2, \ldots, x_k^1, x_k^2) : ((x_1^1, x_1^2), \ldots, (x_k^1, x_k^2)) \in \theta\}.$$

Lemma 6.3. *Let Γ_2 be a finite set of relations on A^2. Let $\Gamma_3 = \{\widetilde{\theta} : \theta \in \Gamma_2\}$. Then there is a positive quantifier-free reduction of* $\mathrm{CSP}(\Gamma_2)$ *to* $\mathrm{CSP}(\Gamma_3)$.

Proof. Let $\sigma = \{R_1, \ldots, R_s\}$ be the type of the structure $\mathbf{\Gamma}_2$ and let $\tau = \{\widetilde{R_1}, \ldots, \widetilde{R_s}\}$ be the type of the structure $\mathbf{\Gamma}_3$. The reduction is straightforward: given a structure $\mathbf{G}$ of type σ, we create two disjoint copies of it, $\mathbf{G}' = \mathbf{G} \times \{0\} \cup \mathbf{G} \times \{1\}$, and define for each $1 \leq i \leq s$

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

It is not difficult to verify that there is a homomorphism from $\mathbf{G}$ to $\mathbf{\Gamma}_2$ if and only if there is one from $\mathbf{G}'$ to $\mathbf{\Gamma}_3$. It remains to show that the reduction is positive and quantifier-free; it is in fact 2-ary with 2 parameters: let

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

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

$$\phi_{\widetilde{\theta}}(x_1^1, x_1^2, \ldots, x_{2r}^1, x_{2r}^2, y^0, y^1) \equiv$$
$$\equiv \left[(x_1^1, x_3^1, \ldots, x_{2r-1}^1) \in \tilde{\theta}\right] \bigwedge \bigwedge_{1 \leq i \leq r} \left[(x_{2i-1}^1 = x_{2i}^1) \bigwedge (x_{2i-1}^2 = y^0) \bigwedge (x_{2i}^2 = y^1)\right].$$

6.2 The "core to idempotent" reduction

Lemma 6.4. *Let Γ_0 be a finite set of relations on A such that $\mathbf{\Gamma}_0$ 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* CSP(Γ_1) *to* CSP(Γ_0).

Proof. Let σ be the signature of $\mathbf{\Gamma}_1$ and let τ be the signature of $\mathbf{\Gamma}_0$. Let $\mathbf{G}_1$ be a structure of type σ. We construct a τ-structure $\mathbf{G}_0$ as follows: it is a copy of $\mathbf{G}_1$ from which we remove all relations in $\Gamma' - \Gamma$, together with a copy of $\mathbf{\Gamma}_0$. 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 $\theta(\mathbf{\Gamma}_0)$ by replacing all occurrences of a by x. Then $\mathbf{G}_1$ admits a homomorphism to $\mathbf{\Gamma}_1$ if and only if $\mathbf{G}_0$ admits a homomorphism to $\mathbf{\Gamma}_0$: indeed one direction is immediate. Suppose now that there is a homomorphism $f : \mathbf{G}_0 \to \mathbf{\Gamma}_0$; we need to prove that if an element $g \in G$ is constrained to $\{a\}$ then $f(g) = a$. Because $\mathbf{\Gamma}_0$ is a core, the restriction of f to the copy of $\mathbf{\Gamma}_0$ in $\mathbf{G}_0$ is an isomorphism, and hence we may find an automorphism α of $\mathbf{\Gamma}'_0$ such that $\alpha \circ f$ is the identity. Consider the substructure of $\mathbf{G}_0$ induced by the copy of $\mathbf{\Gamma}_0$, minus the element a plus the element g: by construction it is isomorphic to the core $\mathbf{\Gamma}_0$, 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 τ in σ. 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 $(y^0, \ldots, y^0, y^1, y^0, \ldots, y^0)$ where y^1 is in the a-th position (the copy of $\mathbf{\Gamma}_0$).

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

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

and

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

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

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

and

$$\phi_\theta^Y(x_1, \ldots, x_r, y^0, y^1) \equiv \bigvee_{(a_1, \ldots, a_r) \in \theta(\mathbf{\Gamma}_0)} \left[\bigwedge_{1 \leq i \leq r} \bigwedge_{j \neq a_i} (x_i^{a_i} = y^1) \wedge (x_i^j = y^0) \right],$$

and

$$\phi_\theta^M(x_1,\dots,x_r,y^0,y^1) \equiv \bigvee_{a\in A} \bigvee_{(a_1,\dots,a_r)\in\theta(\mathbf{\Gamma}_0)} \left[\bigwedge_{i:a_i=a} \phi_U^D(x_i^1,\dots,x_i^h,y^0,y^1) \wedge (x_i^1 \in \{a\})\right] \wedge$$
$$\wedge \left[\bigwedge_{i:a_i\neq a} \left(\bigwedge_{j\neq a_i} (x_i^{a_i} = y^1) \wedge (x_i^j = y^0)\right)\right].$$

6.3 The reductions for "inferred" constraints

Let Γ be a set of relations and suppose that $Pol(\Gamma) \subseteq Pol(\theta)$. We shall require the following combinatorial descriptions of the relation θ in terms of those in Γ (see e.g. [4]).

Lemma 6.5. *Let Γ and Γ' be sets of relations on A. Then the following conditions are equivalent:*

1. *$Pol(\Gamma) \subseteq Pol(\Gamma')$;*
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 ψ 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 Γ_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 first 5 constructions described above induce positive first-order reductions. The 6th one, adding the equality, although not first-order, does preserve expressibility in all restrictions of Datalog we use, see [8]. See also [17] Lemma 3.1 for the proof that all 6 are indeed reductions.

Lemma 6.6. *Let Γ 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* CSP(Γ') *to* CSP(Γ).

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

Lemma 6.7. *Let Γ be a finite set of relations on A, let $\theta \in \Gamma$ be r-ary. Let $\Gamma' = \Gamma \cup \{\theta_1\}$ where $\theta_1 = \{(x_{\pi(1)}, \ldots, x_{\pi(r)}) : (x_1, \ldots, x_r) \in \theta\}$ for some permutation π of $\{1, \ldots, k\}$. Then there is a 1-ary, 0-parameter positive quantifier-free reduction of* $\mathrm{CSP}(\Gamma')$ *to* $\mathrm{CSP}(\Gamma)$.

Proof. Given a structure $\mathbf{G}$ let $\mathbf{G}'$ be obtained from $\mathbf{G}$ by removing θ_1 but adding to $\theta(\mathbf{G}')$ all appropriate permutations of tuples in $\theta_1(\mathbf{G})$. Formally, the interpretation is defined as follows: $\phi_U \equiv TRUE$ and for every k-ary $\rho \neq \theta$ in the signature of $\mathbf{\Gamma}$, let $\phi_\rho \equiv (x_1, \ldots, x_k) \in \rho$ and define

$$\phi_\theta(x_1, \ldots, x_r) \equiv (x_1, \ldots, x_r) \in \theta \bigvee (x_{\pi(1)}, \ldots, x_{\pi(r)}) \in \theta_1.$$

Lemma 6.8. *Let Γ 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* $\mathrm{CSP}(\Gamma')$ *to* $\mathrm{CSP}(\Gamma)$.

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

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

and

$$\phi_\beta(x_1, \ldots, x_r) \equiv (x_1, \ldots, x_r) \in \beta \bigvee (x_1, \ldots, x_r) \in \gamma.$$

Lemma 6.9. *Let Γ be a finite set of relations on A, let $\alpha, \beta \in \Gamma$ be k- and r-ary respectively. Let $\gamma = \{(x_1, \ldots, x_k, y_1, \ldots, y_r) : \overline{x} \in \alpha, \overline{y} \in \beta\}$, and let $\Gamma' = \Gamma \cup \{\gamma\}$. Then there is a 1-ary, 0-parameter positive first-order reduction of* $\mathrm{CSP}(\Gamma')$ *to* $\mathrm{CSP}(\Gamma)$.

Proof. Given a structure $\mathbf{G}$ let $\mathbf{G}'$ be obtained from $\mathbf{G}$ by removing γ 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: $\phi_U \equiv TRUE$ and for every k-ary $\rho \neq \alpha, \beta$ in the signature of $\mathbf{\Gamma}$, let $\phi_\rho \equiv (x_1, \ldots, x_k) \in \rho$, and

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

and

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

Lemma 6.10. *Let Γ be a finite set of relations on A, let $\theta \in \Gamma$ be k–ary and let $\theta_1 = \{(x_1, \ldots, x_{k-1}) : \exists x_k\, \overline{x} \in \theta\}$. Let $\Gamma' = \Gamma \cup \{\theta_1\}$. Then there is a $(k+1)$-ary, 2-parameter positive quantifier-free reduction of* $\mathrm{CSP}(\Gamma')$ *to* $\mathrm{CSP}(\Gamma)$.

Proof. Given a structure $\mathbf{G}$, let $\mathbf{G}'$ be obtained from $\mathbf{G}$ by removing θ_1, and adding to its universe, for each tuple $t = (x_1, \ldots, x_{k-1})$ of $\theta(\mathbf{G})$ a new element x_t, and adding to $\theta(\mathbf{G}')$ the tuple $(x_1, \ldots, x_{k-1}, x_t)$.

Formally the interpretation is defined as follows. Let τ be the type of $\mathbf{\Gamma}$ and let σ be the type of $\mathbf{\Gamma}'$. Define $\phi_U(x^1, \ldots, x^{k+1}, y^0, y^1)$ to be the formula $\phi_U^D \vee \phi_U^E$ where

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

and

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

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 $(x^1, \ldots, x^{k-1}, y^0, y^1)$.

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

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

Finally define

$$\phi_\theta(x_1, \ldots, x_k, y^0, y^1) \equiv \bigwedge_{1 \leq i \leq k-1} \phi_U^D(x_i^1, \ldots, x_i^{k+1}, y^0, y^1) \bigwedge (x_1^1, \ldots, x_{k-1}^1) \in \theta_1 \bigwedge \bigwedge_{1 \leq j \leq k-1} (x_k^j = x_j^1) \bigwedge (x_k^k = y^0) \bigwedge (x_k^{k+1} = y^1)$$

7 Preserving Datalog and its restrictions

In what follows we shall require the following definition (see [5]). Let σ be a vocabulary, and let $S_1, \ldots, S_l$ be relation symbols. Consider a second-order formula

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

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

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

where ϕ is a disjunct of atomic formulas over σ (we also allow equalities) and Σ is a disjunct of atomic formulas involving only the S_i (we allow Σ to be "empty", i.e. FALSE.)

We say that Ψ is *restricted Krom* if each Σ 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) $\overline{x} \in S_i$, (iii) $\neg(\overline{x} \in S_j)$ or (iv) $(\overline{x} \in S_i) \vee \neg(\overline{y} \in S_j)$. We say that Ψ is *monotone* if every disjunct of every ϕ is negated, i.e. each ϕ is of the form

$$\phi \equiv \neg(\overline{x} \in \theta) \vee \cdots$$

where the θ are relational symbols in σ. We say that Ψ is *symmetric* if, whenever the disjunct

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

appears then its "symmetric" also appears as a conjunct, namely

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

We say that a class of σ-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 ϕ to be of the form $\neg(x = y)$ or also $(x = y)$ (of course equality is interpreted normally in each structure).

The following result is an adaptation of Lemma 1 in [5], with parts of the proof of Lemma 3 of [10] peppered in (consult [8] for a detailed proof of the equivalence of the first two conditions). Datalog($\neq$)is defined as follows: it is the same as Datalog, but we also allow equalities and disequalities as EDB's (see [14].) 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 7.1. *Let $\mathcal{C}$ be a collection of σ-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 [5]. 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 $\phi_q = TRUE$, we can simply remove it, and if some $\phi_q = FALSE$, 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 [10]: 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 [10] that P' is actually a program for $\mathcal{C}$.

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 [5], (see also [8]), we obtain that if a class of structure is definable in monotone, restricted, (symmetric), Krom SNP, then we may assume that in the defining Krom formula, equalities are never used.

Proposition 7.1. *Let $\mathcal{C}$ be a class of structures of signature σ and let $\mathcal{D}$ be a class of structures of signature τ. 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 σ-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 Ψ defining it, of the following form:

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

where ψ is

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

where for each $t \in T$, ϕ_t is a disjunct of negated atomic formulas over τ, and each Σ_t is a disjunct of at most one positive and one negated atom with symbols in $\{S_1, \dots, S_l\}$. As we noted earlier, we may assume that all atomic formulas in the ϕ'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 ϕ_U, ϕ_θ for each $\theta \in \tau$. Consider the following procedure (inspired by Lemma 1 of [2]): our new formula Ψ' 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 ϕ_θ of our reduction appear in this list, and furthermore we assume that no variable u^i appears in more than one ϕ_θ. It will also have second-order variables $S_1', \dots, S_l'$.

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

– if ϕ_θ is quantifier-free, let

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

– if $\phi_\theta \equiv \exists u_{i_1} \ldots u_{i_w} \mu_\theta$ where μ_θ is quantifier-free, let

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

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

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

$$\mathcal{I}(\delta) \equiv (v_1^{i_1}, \ldots, v_k^{i_1}, \ldots, v_1^{i_r}, \ldots, v_k^{i_r}) \in S_i';$$

for each t let Σ_t' denote the result of this substitution on the disjuncts of Σ_t.

Furthermore, define the following σ-formulas: let

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

and let

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

Now we define Ψ' as follows:

$$\Psi' \equiv \exists S_1' \ldots S_l' \forall v_1^1 \ldots v_k^1, \ldots, v_1^m \ldots v_k^m \forall y^1 \ldots y^p \forall u^1 \ldots u^s \, \psi'$$

where ψ' is

$$\psi' \equiv \bigwedge_T \left(\mathcal{I}(\phi_t) \vee Y \vee Z \vee \Sigma_t' \right).$$

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 σ-structure with less than p points satisfies Ψ', 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. *Ψ' is restricted, monotone Krom with equalities. It is symmetric if Ψ is.*

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

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

where the δ_i are atoms. By the hypothesis on our reductions and by construction, each $\mathcal{I}(\delta_i)$ is a positive Boolean combination of atoms (atoms 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 atoms. 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 ε_{kl} are negated atoms. Now it is clear that our formula is monotone, restricted Krom with equalities.

For symmetry, notice that any conjunct of Ψ'

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

is obtained from some conjunct

$$\phi_t \vee \Sigma_t$$

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

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

appears in Ψ' (obviously the symmetric of Σ_t' is Σ_u' if $\overline{\Sigma_t} = \Sigma_u$).

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

Proof of Claim 2. By definition of our reductions the choice of c is immaterial, i.e. $\mathcal{I}(G, c)$ satisfies Ψ 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 a r-ary relation on $\mathcal{I}(G, c) \subseteq G^k$), i.e.

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

precisely if

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

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

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

of Φ': we must show that for this choice of values this conjunct is satisfied. Suppose then that none of Y, Z nor Σ_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 $c = (y^1, \ldots, y^p)$ is proper. Hence $\mathcal{I}(G, c)$ satisfies Ψ for $c = (y^1, \ldots, y^p)$.

2. Since our values do not satisfy Z, it means that each tuple $v^i = (v^i_1, \ldots, v^i_k)$ is in the universe of $\mathcal{I}(G, c)$ (i.e. satisfies ϕ_U.)

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

$$\phi_t \vee \Sigma_t$$

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

There is some index set Q such that

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

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

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

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

$$\phi_\theta \equiv \exists u_{i_1} \ldots u_{i_w} \mu_\theta$$

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

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

Proof of Claim 3. Choose some proper $c = (z_1, \ldots, z_p)$. 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.

$$((x^1_1, \ldots, x^1_k), \ldots, (x^r_1, \ldots, x^r_k)) \in S_i$$

if and only if

$$(x^1_1, \ldots, x^1_k, \ldots, x^r_1, \ldots, x^r_k) \in S'_i$$

and $\phi_U(x^i_1, \ldots, x^i_k, y^1, \ldots, y^p)$ holds for all $1 \leq i \leq m$.

Choose some elements $v^i = (v^i_1, \ldots, v^i_k)$ in the universe of $\mathcal{I}(G, c)$ (i.e. they all satisfy ϕ_U), and choose some conjunct $\phi_t \vee \Sigma_t$ of Ψ. Suppose that Σ_t does not hold: we must show that ϕ_t does (for our choice of v^i's). Now by definition of the S_i and Σ'_t it is immediate that Σ'_t isn't satisfied either. Hence $Y \vee Z \vee \mathcal{I}(\phi_t)$ holds for any choice of $u^1, \ldots, u^s$. 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 $u^1, \ldots, u^s$. We will conclude that ϕ_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 γ_q is an atomic formula with relational symbol in τ. Suppose for a contradiction that every γ_q holds in $\mathcal{I}(G, c)$. Choose some $q \in Q$ and suppose first that

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

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

$$\phi_\theta \equiv \exists u_{i_1} \ldots u_{i_w} \mu_\theta$$

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

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 $\mathcal{I}(G, c) \in \mathcal{D}$ iff $\mathcal{I}(G, c)$ satisfies Ψ iff $\mathbf{G}$ satisfies Ψ'.

Corollary 7.1. *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 7.1, 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 7.1 we conclude that for some $p \geq 0$, the set consisting of all σ-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 7.1 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

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

where $P()$ is the goal predicate of the original program, $x_1, \ldots, x_k$ represent the elements of G and ϕ 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 7.1 one last time to conclude that it is definable in linear (symmetric) Datalog.

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

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

1. *Let $\mathbb{C}$ be a finite algebra in $\mathcal{V}(\mathbb{A})$, and let Γ_0 be a finite set of relations invariant under the basic operations of $\mathbb{C}$. Then there exists a logspace many-one reduction of* CSP(Γ_0) *to* CSP(Γ)*. Furthermore, if $\neg CSP(\Gamma)$ is expressible in (linear, symmetric) Datalog, then so is $\neg CSP(\Gamma_0)$.*
2. *If furthermore $\mathbb{C} \in HS(\mathbb{A})$ and the relations in Γ_0 are irredundant, then the above reduction is first-order.*

Proof. (1) We shall define a sequence of CSP's 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 [17]. By hypothesis on the algebra $\mathbb{C}$ there exist an integer m, a subalgebra $\mathbb{B}$ of $\mathbb{A}^m$ and a surjective homomorphism $\mu : \mathbb{B} \to \mathbb{C}$.

Let $\Gamma_1 = \{\mu^{-1}(\theta) : \theta \in \Gamma_0\}$. By Lemma 6.1 there is a first-order reduction from CSP(Γ_0) to CSP(Γ_1) which by Corollary 7.1 preserves expressibility in (linear, symmetric) Datalog. Furthermore, it is easy to see that all relations in Γ_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 6.2 we have a first-order reduction of CSP(Γ_1) to CSP(Γ_2) which preserves expressibility in (linear, symmetric) Datalog by Corollary 7.1. Furthermore, it is easy to see that all relations in Γ_2 are invariant under the operations of $\mathbb{A}^m$.

Now define $\Gamma_3 = \{\widetilde{\theta} : \theta \in \Gamma_2\}$ (see definition just before Lemma 6.3.) Once again we obtain a first-order reduction from CSP(Γ_2) to CSP(Γ_3) preserving expressibility in restrictions of Datalog by Lemma 6.3 and Corollary 7.1.

It is easy to see that the relations in Γ_3 are invariant under the basic operations of $\mathbb{A}$. Since these are precisely the idempotent operations that preserve all relations in Γ, we obtain that $Pol(\Gamma') \subseteq Pol(\Gamma_3)$ where $\Gamma' = \Gamma \cup \{\{a\} : a \in A\}$. There exists a finite sequence of sets $\Gamma' = \Lambda_0, \ldots, \Lambda_s = \Gamma_3$ such that each set is obtained from the previous by one of the 6 constructions described in Lemma 6.5 (3). By Lemmas 6.6 to 6.10 and Corollary 7.1, 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 in [13], and also preserves expressibility by [8]. Hence we have a logspace reduction of CSP(Γ_3) to CSP(Γ'), that preserves expressibility.

Finally we invoke Lemma 6.4 to obtain a first-order reduction of CSP(Γ') to CSP(Γ) which preserves expressibility by Corollary 7.1.

(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 Γ_0 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 Γ_0 are. Now we show that in the sequence of sets Λ_i we never need to add the equality relation. Indeed, by part (2) of Lemma 6.5 for every $\gamma \in \Gamma_3$ there exists a primitive positive formula defining it in terms of the relations in Γ'. We claim that if γ 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 γ 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 $\overline{x} \in \theta_j$ for relations $\theta_j \in \Gamma$. So let's say that $x_1, \ldots, x_h$ is a list of all distinct variables appearing in the formula. For each conjunct $(x_{i_1}, \ldots, x_{i_s}) \in \theta_j$ the relation

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

is obtained by a permutation of the product $A^{h-s} \times \theta_j$.[1] Finally, our relation is the intersection of all these relations.

8 The Reduction in Theorem 3.1 is First-Order

We described in the proof of Theorem 3.1 a reduction from NOT st-connectivity to CSP(Γ') where $\Gamma' = \Gamma \cup \{\{a\} : a \in A\}$ and CSP(Γ) is not first-order definable. We complete the proof of the theorem by showing that the reduction is indeed FO.

Proof. 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 $\mathcal{I}(\mathbf{G}) = \mathbf{K}$ is the subset of $G \times G^m$ consisting of all tuples $(g; x^1, \ldots, x^m)$ such that $g \in G$ and $(x^1, \ldots, x^m)$ 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.

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

- 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

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

such that

1. there exists $x, y \in G$ such that xEy and $\{g_1, \ldots, g_a\} \subseteq \{x, y\}^a$, and
2. $(y_1, \ldots, y_a) \in R_i(\mathbf{\Gamma}^2)$ where y_j is the element of A^2 indexed by $(x_j^1, \ldots, x_j^m)$ 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 can we can list all possible subsets of $\{x, y\}^a$ and so the formula will be of the form

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

- 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 all tuples $(g; x^1, \ldots, x^m)$ such that either

[1] Of course, here we are assuming the obvious fact that we may add the unary relation A to any CSP with base set A without changing anything.

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

9 2 element case

We give a detailed proof of Theorem 5.1:

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

1. *If $\mathcal{V}$ admits the unary type then $CSP(\Gamma)$ is NP complete, and $\neg CSP(\Gamma)$ is not expressible in Datalog.*
2. *If $\mathcal{V}$ omits the unary type but admits the affine type, then $CSP(\Gamma)$ is $\oplus L$ complete and $\neg CSP(\Gamma)$ is not expressible in Datalog.*
3. *if $\mathcal{V}$ omits the unary and affine types, but admits the semilattice type, then $CSP(\Gamma)$ is P complete; if Γ* is finite *then $\neg CSP(\Gamma)$ is expressible in Datalog, but not in linear Datalog.*
4. *if $\mathcal{V}$ omits the unary, affine and semilattice types, but admits the lattice type, then $CSP(\Gamma)$ is NL complete; if Γ is finite, then $\neg CSP(\Gamma)$ is expressible in linear Datalog, but not in symmetric Datalog;*
5. *if $\mathcal{V}$ admits only the Boolean type, then $CSP(\Gamma)$ is either first-order definable or L complete; if Γ is finite, then $\neg CSP(\Gamma)$ is expressible in symmetric 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 [1]: all references to special operations on $\{0,1\}$ and clones in Post's lattice use the notation from that reference.

Fix Γ a (not necessarily finite) set of relations on $\{0,1\}$ such that $\mathbf{\Gamma}$ is a core, i.e. the clone $Pol(\Gamma)$ contains no constant operation. Let $\mathbb{A}$ denote the 2-element algebra associated to the problem $CSP(\Gamma)$, and let $\mathcal{V}$ denote the variety generated by $\mathbb{A}$. We consider the possibilities for the clone of terms $Pol(\Gamma)$ 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 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 [8].

Now suppose $\mathbb{A}$ is idempotent. Clearly $\mathbb{A}$ is strictly simple and hence we may use Lemma 1.1 and Corollary 1.1 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 $CSP(\Gamma)$ 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 $CSP(\Gamma)$ 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 $CSP(\Gamma)$ is P complete. If Γ is finite, the problem is HORN k-SAT which is in $(1, k)$-Datalog [9].

- 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 $CSP(\Gamma)$ 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 [5] since it contains a majority operation. As for S_{10}, the argument is as follows: (1) if we take Γ finite such that $Pol(\Gamma)$ contains S_{10} then $Pol(\Gamma)$ is above some S_{10}^n and thus contains the near-unanimity operation h_n; then expressibility in linear Datalog follows from [5].

- Suppose that $\mathcal{V}$ omits all types but the Boolean type. It follows from [19] 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 $CSP(\Gamma)$ is thus L complete or first-order definable.
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

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

We claim that the clones in question are the S_{12}^k and that in fact for each k we have $S_{12}^k = Pol(R_k)$. Indeed it is trivial to verify that the operation

$x \wedge (y \vee \overline{z})$ preserves each R_k and hence

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

so obviously $Pol(R_k)$ 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 $S_{12}^k \neq Pol(R_{k+1})$. On the other hand, a simple application of the pigeonhole principle shows that h_k preserves R_k. It follows that the problem $CSP(\Gamma)$ 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 $CSP(\Gamma)$ is L-complete.

If Γ is finite, then we are in case (1) or (2) and expressibility in symmetric Datalog follows from [8].

10 Mod$_p$L Hardness

In this section we outline the proof that, if $\mathcal{V}(\mathbb{A}(\Gamma))$ admits the unary or affine type then there exists a prime p such that CSP(Γ) is Mod$_p$L-hard under FO reductions. Recall that Mod$_p$L is the class of languages recognized by Mod$_p$-counting non-deterministic logspace machines. Formally, $K \in$ Mod$_p$L is 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 $\oplus$L.

These classes contain a number of natural problems related to modular arithmetic such as solving systems of linear equations over $\mathbb{Z}_p$ (see e.g. Buntrock, Damm, Hertrampf and Meinel: *Structure and importance of logspace-MOD classes. Mathematical Systems Theory*, **25**(3):223–237, 1992.). In particular, it can be shown from their work that solving systems of linear equations modulo p^k is also in Mod$_p$L and so the CSP(Γ) analyzed for the affine case of Theorem 5.2 is indeed Mod$_p$L-complete.

Let p be any prime number. We denote by $NOT\ p-GAP$ the set of all digraphs with two distinguished nodes s and t such that the number of 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 **G** with edge relation $\theta(G)$ we may "remove" all loops by defining a new digraph G' on the same universe with edge relation $\theta(G')$ 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.

Lemma 10.1. *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 NC^1 reductions. The reduction can be made FO using a straightforward adaptation of the NL-completeness of REACH under FO reductions. (Theorem 3.16 of Immerman's book)

Lemma 10.2. *Let $\mathbb{A} = \langle A; +, 0\rangle$ be a finite Abelian group with at least 2 elements. Consider the set Γ that consists of (i) the 3-ary relation $\mu = \{(x, y, z) \in A^3 : x + y = z\}$, (ii) the unary relation $B = \{b\}$ where b is some non-zero element of $\mathbb{A}$ such that $pb = 0$ for some prime p, and (iii) the unary relation $Z = \{0\}$. Then $\mathrm{CSP}(\Gamma)$ is mod-p L hard under first-order reductions.*

Proof. By the last result, it will suffice to find a first-order reduction of *NOT* $p-GAP$ to $\mathrm{CSP}(\Gamma)$. The basic idea is rather straightforward: Let $\mathbf{G}$ be a digraph with specified vertices s and t. Recall that (a) we may assume all our digraphs have no loops, and (b) we may access the ordering relation on $\mathbf{G}$, which has the property that edges are monotone and $1 = s$ and $n = t$. 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 goes 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 goes 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 τ 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

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

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

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

where

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

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.]

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/