Solving Order Constraints in Logarithmic Space

Andrei Krokhin[1]* and Benoit Larose[2]**

[1] Department of Computer Science, University of Warwick
Coventry, CV4 7AL, UK
andrei.krokhin@dcs.warwick.ac.uk
[2] Champlain Regional College,
900 Riverside Drive, St-Lambert, Québec, Canada J4P 3P2, and
Department of Mathematics and Statistics, Concordia University
1455 de Maisonneuve West, Montréal, Québec, Canada H3G 1M8
larose@mathstat.concordia.ca

Abstract. We combine methods of order theory, finite model theory, and universal algebra to study, within the constraint satisfaction framework, the complexity of some well-known combinatorial problems connected with a finite poset. We identify some conditions on a poset which guarantee solvability of the problems in (deterministic, symmetric, or non-deterministic) logarithmic space. On the example of order constraints we study how a certain algebraic invariance property is related to solvability of a constraint satisfaction problem in non-deterministic logarithmic space.

1 Introduction

A wide range of combinatorial search problems encountered in artificial intelligence and computer science can be naturally expressed as 'constraint satisfaction problems' [7, 35], in which the aim is to find an assignment of values to a given set of variables subject to specified constraints. For example, the standard propositional satisfiability problem [37, 41] may be viewed as a constraint satisfaction problem where the variables must be assigned Boolean values, and the constraints are specified by clauses. Further examples include graph colorability, clique, and Hamiltonian circuit problems, conjunctive-query containment, and many others (see [20, 26]). One advantage of considering a common framework for all of these diverse problems is that it makes it possible to obtain generic structural results concerning the computational complexity of constraint satisfaction problems that can be applied in many different areas such as machine vision, belief maintenance, database theory, temporal reasoning, type reconstruction, graph theory, and scheduling (see, e.g., [12, 20, 26, 27, 35, 42]).

The general constraint satisfaction problem (CSP) is **NP**-complete. Therefore, starting with the seminal paper by Schaefer [41], much work has been done

on identifying restrictions on the problem that guarantee lower complexity (see, e.g., [3–5, 7–9, 12–14, 20–24, 26]).

The constraint satisfaction problem has several equivalent definitions (see, e.g., [5, 7, 14, 20, 26]). For the purposes of this paper, we define it, as in [8, 12, 20, 26], as the homomorphism problem: given two relational structures, $\mathcal{A}$ and $\mathcal{B}$, the question is whether there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. One distinguishes uniform and non-uniform CSPs depending on whether both $\mathcal{A}$ and $\mathcal{B}$ are parts of the input or not (see, e.g., [26]). The case when the choice of $\mathcal{A}$ is restricted has been studied in connection with database theory (see, e.g., [13, 14, 16, 26]). In this paper, as in [8, 12, 20], we will consider non-uniform constraint satisfaction problems CSP($\mathcal{B}$) where the structure $\mathcal{B}$ is fixed.

We will concentrate on the much understudied case of constraint problems in logarithmic space complexity classes. The class **NL** of problems solvable in non-deterministic logarithmic space has received much attention in complexity theory. It is known that **NL** $\subseteq$ **NC**, and so problems in **NL** are highly paralellizable.

Despite the large amount of tractable constraint satisfaction problems identified so far, to the best of our knowledge, only two families of concrete problems CSP($\mathcal{B}$) are known to be in **NL**. The first family consists of restricted versions of the Boolean satisfiability problem: the so-called bijunctive constraints [41], including the 2-SATISFIABILITY problem, and implicative Hitting-Set Bounded constraints [7, 8]. The second family consists of implicational constraints [24] which are a generalization of bijunctive constraints to non-Boolean problems, and which, in turn, have been slightly generalized in [8]. Dalmau introduced 'bounded path duality', a general sufficient condition for a CSP($\mathcal{B}$) to be in **NL**, which is shown to be satisfied by all problems from the two families [8]. However, this condition is not easy to apply; in general, it is not known to be decidable.

Our main motivation in this paper is to clarify the relation between solvability of a CSP in non-deterministic logarithmic space and the algebraic condition of invariance under a near-unanimity operation that is also satisfied by all concrete problems above. It is known that many important properties of CSPs can be captured by algebraic invariance conditions (see, e.g., [9, 12, 20]), and that the condition mentioned above exactly corresponds to the following property: for some l, every instance of CSP($\mathcal{B}$) has precisely the same solutions as the system of all its subinstances of size l [12, 21]. It is unknown whether this property is sufficient and/or necessary for a CSP($\mathcal{B}$) to be in **NL**. We answer this question for problems connected with posets by showing that, for such problems, the condition is sufficient but not necessary (and therefore it is not necessary in general).

In order theory, there is a rich tradition of linking combinatorial and algebraic properties of posets (see, e.g., [6, 28–30, 39, 45, 46]). We use such results in order to get combinatorial properties of posets that can be described in first-order logic with the transitive closure operator. As is known from finite model theory, sentences in this logic can be evaluated in non-deterministic logarithmic space. We use the poset retraction problem as a medium for applying finite

model theory results to the considered CSPs. The notion of retraction plays an important role in order theory (see, e.g., [11, 28, 39]), and this problem has also been studied in computer science along with the problem of satisfiability of inequalities in a poset and the extendibility problem (see, e.g., [2, 12, 29, 38]). As intermediate results that are of independent interest, we describe two classes of posets for which the retraction problem is solvable in deterministic or (complete for) symmetric logarithmic space.

The complexity class **SL** (symmetric log-space) appears in the literature less often than **L** and **NL**, so we say a few words about it here. It is the class of problems solvable by symmetric non-deterministic Turing machines in logarithmic space [31], and **L** $\subseteq$ **SL** $\subseteq$ **NL**. This class can be characterized in a number of ways (see, e.g., [1, 15, 36]) and is known to be closed under complementation [36]. It contains such important problems as the UNDIRECTED st-CONNECTIVITY problem and the GRAPH 2-COLORABILITY problem [1, 36, 40]; it was shown in [25] that the GRAPH ISOMORPHISM problem restricted to colored graphs with color multiplicities 2 and 3 is **SL**-complete. It is known (see, e.g., [15]) that sentences of **FO** + **STC**, first-order logic with the symmetric transitive (reflexive) closure operator, can be evaluated in symmetric logarithmic space.

Proofs of all results in this paper are omitted due to space constraints.

2 Constraint Satisfaction Problems

A *vocabulary* τ is a finite set of relation symbols or predicates. Every relation symbol $R \in \tau$ has an *arity* $ar(R) \geq 0$. A τ-structure $\mathcal{A}$ consists of a set A, called the *universe* of $\mathcal{A}$, and a relation $R^{\mathcal{A}}$ of arity $ar(R)$ for every $R \in \tau$. All structures considered in this paper are *finite*.

A *homomorphism* from a τ-structure $\mathcal{A}$ to a τ-structure $\mathcal{B}$ is a mapping $h : A \to B$ such that $(a_1, \ldots, a_{ar(R)}) \in R^{\mathcal{A}}$ implies $(h(a_1), \ldots, h(a_{ar(R)})) \in R^{\mathcal{B}}$ for every $R \in \tau$. If there exists a homomorphism from $\mathcal{A}$ to $\mathcal{B}$, we write $\mathcal{A} \to \mathcal{B}$.

We consider the problem CSP($\mathcal{B}$) where only structure $\mathcal{A}$ is the input, and the question is whether $\mathcal{A} \to \mathcal{B}$. Of course, one can view CSP($\mathcal{B}$) as the class of all τ-structures $\mathcal{A}$ such that $\mathcal{A} \to \mathcal{B}$. So we have a class of problems parameterized by finite structures $\mathcal{B}$, and the ultimate goal of our research is to classify the complexity of such problems. The classic problems of this type are various versions of GRAPH COLORABILITY and SATISFIABILITY of logical formulas.

Example 1 (H-coloring). If $\mathcal{B}$ is an undirected irreflexive graph H then CSP($\mathcal{B}$) is the GRAPH H-COLORING problem. This problem is **SL**-complete if H is bipartite [36, 40], and **NP**-complete otherwise [17]. If H is a complete graph K_k then CSP($\mathcal{B}$) is the GRAPH k-COLORABILITY problem.

Example 2 (2-SAT). Let $\mathcal{B}$ be the stucture with universe $\{0, 1\}$ and all at most binary relations over $\{0, 1\}$. Then CSP($\mathcal{B}$) is exactly the 2-SATISFIABILITY problem, it is **NL**-complete [37].

Example 3 (NAE-SAT). Let $\mathcal{B}$ be the stucture with universe $\{0,1\}$ and one ternary relation $R = \{(a,b,c) \mid \{a,b,c\} = \{0,1\}\}$. Then $\mathrm{CSP}(\mathcal{B})$ is exactly the NOT-ALL-EQUAL SATISFIABILITY problem as defined in [41], it is **NP**-complete.

The most significant progress in classifying the complexity of $\mathrm{CSP}(\mathcal{B})$ has been made via methods of finite model theory [12, 26] and methods of universal algebra [3–5, 20–23]. In [8, 9], both approaches are present.

The finite model theory approach aims at defining $\mathrm{CSP}(\mathcal{B})$ or its complement in various logics. Most of the known results in this direction make extensive use of the logic programming language Datalog.

A *Datalog program* over a vocabulary τ is a finite collection of rules of the form

$$t_0 : -t_1, \ldots, t_m$$

where each t_i is an atomic formula $R(v_1, \ldots, v_l)$. The predicates occuring in the heads of the rules are the *intensional database* predicates (IDBs), while all others are *extensional database* predicates (EDBs) and must belong to τ. One of the IDBs is designated as the goal predicate. A Datalog program is a recursive specification of the IDBs with semantics obtained via least fixed-points (see [43]).

For $0 \leq j \leq k$, (j,k)-Datalog is the collection of Datalog programs with at most k variables per rule and at most j variables per rule head. A Datalog program is *linear* if every rule has at most one IDB in its body.

In [12], tractability of many constraint satisfaction problems was explained in the following way: if $\neg\mathrm{CSP}(\mathcal{B})$, the complement of $\mathrm{CSP}(\mathcal{B})$, is definable in (j,k)-Datalog for some j,k then $\mathrm{CSP}(\mathcal{B})$ is solvable in polynomial time. Dalmau [8] introduced "bounded path duality" for $\mathcal{B}$, a general sufficient condition for $\mathrm{CSP}(\mathcal{B})$ to be in **NL**. This condition is characterized in seven equivalent ways, one of them being definability of $\neg\mathrm{CSP}(\mathcal{B})$ in linear Datalog. However, only definability by linear $(1,k)$-Datalog programs is known to be decidable (for any fixed k) [8]. It is noted in that paper that all known concrete structures $\mathcal{B}$ with $\mathrm{CSP}(\mathcal{B})$ in **NL** have bounded path duality.

The algebraic approach to $\mathrm{CSP}(\mathcal{B})$ uses the notion of a polymorphism.

Definition 1 *Let R be an m-ary relation on B, and f an n-ary operation on B. Then f is said to be a* polymorphism *of R (or R is* invariant *under f) if, for every $(b_{11}, \ldots, b_{m1}), \ldots, (b_{1n}, \ldots, b_{mn}) \in R$, we have*

$$f \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} = \begin{pmatrix} f(b_{11}, b_{12}, \ldots, b_{1n}) \\ f(b_{21}, b_{22}, \ldots, b_{2n}) \\ \vdots \\ f(b_{m1}, b_{m2}, \ldots, b_{mn}) \end{pmatrix} \in R.$$

An operation on B is said to be a polymorphism of a τ-structure $\mathcal{B}$ if it is a polymorphism of every $R^{\mathcal{B}}$, $R \in \tau$.

Example 4. 1) It is easy to check that both permutations on $\{0, 1\}$ are polymorphisms of the relation R from Example 3, while the binary operation min is not.

2) The polymorphisms of a poset are simply the monotone operations on it.

3) The *dual discriminator* on B is defined by

$$\mu_B(x, y, z) = \begin{cases} y & \text{if } y = z, \\ x & \text{otherwise.} \end{cases}$$

One can verify that $\mu_{\{0,1\}}$ is a polymorphism of the structure $\mathcal{B}$ from Example 2.

Let $\mathrm{Pol}(\mathcal{B})$ denote the set of all polymorphisms of $\mathcal{B}$. This set determines the complexity of $\mathrm{CSP}(\mathcal{B})$, as the following result shows.

Theorem 1 ([20]) *Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be structures with the same universe B. If $\mathrm{Pol}(\mathcal{B}_1) \subseteq \mathrm{Pol}(\mathcal{B}_2)$ then $\mathrm{CSP}(\mathcal{B}_2)$ is polynomial-time reducible to $\mathrm{CSP}(\mathcal{B}_1)$.*

Moreover, if the equality relation $=_B$ on B can be expressed by using predicates of $\mathcal{B}_1$, conjunction, and existential quantification, then the above reduction is logarithmic-space.

Examples of classifying the complexity of $\mathrm{CSP}(\mathcal{B})$ by particular types of polymorphisms can be found in [3–5, 8, 9, 20–23]. It follows from [22, 24] that if μ_B is a polymorphism of $\mathcal{B}$ then $\mathrm{CSP}(\mathcal{B})$ is in **NL**. This result is generalized in [8] to give a slightly more general form of ternary polymorphism guaranteeing that $\mathrm{CSP}(\mathcal{B})$ is in **NL**.

Definition 2 *Let f be an n-ary operation. Then f is said to be* idempotent *if $f(x, \ldots, x) = x$ for all x, and it is said to be* Taylor *if, in addition, it satisfies n identities of the form*

$$f(x_{i1}, \ldots, x_{in}) = f(y_{i1}, \ldots, y_{in}), \quad i = 1, \ldots, n$$

where $x_{ij}, y_{ij} \in \{x, y\}$ for all i, j and $x_{ii} \neq y_{ii}$ for $i = 1, \ldots, n$.

For example, it is easy to check that any binary idempotent commutative operation and the dual discriminator are Taylor operations. The following result links hard problems with the absence of Taylor polymorphisms.

Theorem 2 ([29]) *Let $\mathcal{B}$ be a structure such that $\mathrm{Pol}(\mathcal{B})$ consists of idempotent operations. If $\mathcal{B}$ has no Taylor polymorphism then $\mathrm{CSP}(\mathcal{B})$ is* **NP**-*complete.*

3 Poset-Related Problems

In this section we introduce the particular type of problems we work with in this paper, the poset retraction problem and the order constraint satisfaction problem. Here and in the following the universes of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{P}$, $\mathcal{Q}$ etc. are denoted by A, B, P, Q etc., respectively.

Let $\mathcal{A}$, $\mathcal{B}$ be structures such that $B \subseteq A$. Recall that a *retraction* from $\mathcal{A}$ onto $\mathcal{B}$ is a homomorphism $h : \mathcal{A} \to \mathcal{B}$ that fixes every element of $\mathcal{B}$, that is $h(b) = b$ for every $b \in B$.

Fix a poset $\mathcal{P}$. An instance of the *poset retraction problem* $\mathrm{PoRet}(\mathcal{P})$ is a poset $\mathcal{Q}$ such that the partial order of $\mathcal{P}$ is contained in the partial order of $\mathcal{Q}$, and the question is whether there is a retraction from $\mathcal{Q}$ onto $\mathcal{P}$. Again, we can view $\mathrm{PoRet}(\mathcal{P})$ as the class of all posets $\mathcal{Q}$ with positive answer to the above question, and then $\neg\mathrm{PoRet}(\mathcal{P})$ is the class of all instances $\mathcal{Q}$ with the negative answer. This problem was studied in [2, 12, 38]. For example, it was proved in [12] that, for every structure $\mathcal{B}$, there exists a poset $\mathcal{P}$ such that $\mathrm{CSP}(\mathcal{B})$ is polynomial-time equivalent to $\mathrm{PoRet}(\mathcal{P})$.

In [38], a poset $\mathcal{P}$ is called *TC-feasible* if $\neg\mathrm{PoRet}(\mathcal{P})$ is definable (in the class of all instances of $\mathrm{PoRet}(\mathcal{P})$) by a sentence in a fragment of $\mathbf{FO} + \mathbf{TC}$, first-order logic with the transitive closure operator, over the vocabulary τ_1 that contains one binary predicate R interpreted as partial order and the constants c_p, $p \in P$, always interpreted as the elements of $\mathcal{P}$. The fragment is defined by the condition that negation and universal quantification are disallowed. Since any sentence in $\mathbf{FO} + \mathbf{TC}$ is verifiable in non-deterministic logarithmic space [15] and $\mathbf{NL} = \mathbf{CoNL}$ [19], TC-feasibility of a poset $\mathcal{P}$ implies that $\mathrm{PoRet}(\mathcal{P})$ is in $\mathbf{NL}$. Note that even if $\neg\mathrm{PoRet}(\mathcal{P})$ is definable by a sentence in full $\mathbf{FO} + \mathbf{TC}$ then, of course, $\mathrm{PoRet}(\mathcal{P})$ is still in $\mathbf{NL}$.

Let $P = \{p_1, \ldots, p_n\}$ and let τ_2 be a vocabulary containing one binary predicate R and n unary predicates $S_1, \ldots, S_n$. We denote by $\mathcal{P}_P$ the τ_2-structure with universe P, $R^{\mathcal{P}_P}$ being the partial order $\leq$ of $\mathcal{P}$, and $S_i^{\mathcal{P}_P} = \{p_i\}$. We call problems of the form $\mathrm{CSP}(\mathcal{P}_P)$ *order constraint satisfaction problems*. Note that if $\mathcal{B}$ is simply a poset then $\mathrm{CSP}(\mathcal{B})$ is trivial because every mapping sending all elements of an instance to a fixed element of $\mathcal{B}$ is a homomorphism. The main difference between $\mathrm{CSP}(\mathcal{P}_P)$ and $\mathrm{PoRet}(\mathcal{P})$ is that an instance of $\mathrm{CSP}(\mathcal{P}_P)$ is not necessarily a poset, but an arbitrary structure over τ_2.

The following theorem will provide us with a way of obtaining order constraint satisfaction problems in $\mathbf{NL}$.

Theorem 3 *Let $\mathcal{P}$ be a poset. If $\neg\mathrm{PoRet}(\mathcal{P})$ is definable (within the class of all posets containing $\mathcal{P}$) by a τ_1-sentence in $\mathbf{FO} + \mathbf{TC}$ then $\mathrm{CSP}(\mathcal{P}_P)$ is in $\mathbf{NL}$.*

By analysing the (proof of the) above theorem one can show that the problems $\mathrm{PoRet}(\mathcal{P})$ and $\mathrm{CSP}(\mathcal{P}_P)$ are polynomial-time equivalent. Invoking the result from [12] mentioned in this section, we get the following statement.

Corollary 1 *For every structure $\mathcal{B}$, there exists a poset $\mathcal{P}$ such that $\mathrm{CSP}(\mathcal{B})$ and $\mathrm{CSP}(\mathcal{P}_P)$ are polynomial-time equivalent.*

It is not known whether the equivalence can be made logarithmic-space, and so the corollary cannot now help us in studying CSPs in $\mathbf{NL}$. We believe that Corollary 1 is especially interesting in view of Theorem 1 because polymorphisms of a poset may be easier to analyse. In particular, note that all polymorphisms of $\mathcal{P}_P$ are idempotent, and so, according to Theorem 2, in order to classify the

complexity of CSP($\mathcal{B}$) up to polynomial-time reductions, it suffices to consider posets with a Taylor polymorphism.

Other poset-related problems studied in the literature are the satisfiability of inequalities, denoted $\mathcal{P}$-SAT, and the extendibility problem Ext($\mathcal{P}$). An instance of $\mathcal{P}$-SAT consists of a system of inequalities involving constants from $\mathcal{P}$ and variables, and the question is whether this system is satisfiable in $\mathcal{P}$. The problem $\mathcal{P}$-SAT plays an important role in type reconstruction (see, e.g., [2, 18, 38]), it was shown to be polynomial-time equivalent to PoRet($\mathcal{P}$) for the same $\mathcal{P}$ [38]. An instance of Ext($\mathcal{P}$) consists of a poset $\mathcal{Q}$ and a partial map f from $\mathcal{Q}$ to $\mathcal{P}$, and the question is whether f extends to a homomorphism from $\mathcal{Q}$ to $\mathcal{P}$. The problem Ext($\mathcal{P}$) was studied in [29], where it was shown to be polynomial-time equivalent to CSP($\mathcal{P}_P$), so all four poset-related problems are polynomial-time equivalent.

It is easy to see that PoRet($\mathcal{P}$) is the restriction of Ext($\mathcal{P}$) to instances where Q contains P and $f = \mathrm{id}_P$ is the identity function on P, while the problem Ext($\mathcal{P}$) can be viewed as the restriction of CSP($\mathcal{P}_P$) to instances where the binary relation is a partial order. Of course, if CSP($\mathcal{P}_P$) is in **NL** then so are the other three problems.

4   Near-unanimity Polymorphisms

In this section we prove that posets with a polymorphism of a certain form, called *near-unanimity* operation, give rise to order constraint satisfaction problems in **NL**.

Definition 3 *A near-unanimity (NU) operation is an l-ary (l ≥ 3) operation satisfying*

$$f(y, x, \ldots, x) = f(x, y, x, \ldots, x) = \cdots = f(x, \ldots, x, y) = x$$

for all x, y.

Near-unanimity operations have attracted much attention in order theory and universal algebra (see, e.g., [10, 28–30, 39, 45, 46]). For example, the posets having a ternary NU polymorphism (known as a *majority* operation) are precisely retracts of direct products of fences [39], where a fence is a poset on $\{a_0, \ldots, a_k\}$ such that $a_0 < a_1 > a_2 < \ldots a_k$ or $a_0 > a_1 < a_2 > \ldots a_k$, and there are no other comparabilities. Posets with an NU polymorphism (of some arity) are characterized in a number of ways in [28, 30].

It was proved in [12, 21] (using different terminology) that if a structure $\mathcal{B}$ has an NU polymorphism then CSP($\mathcal{B}$) is solvable in polynomial time. In [12], such problems are said to have *bounded strict width*, while in [21] they are shown to be related to the so-called *strong consistency*, a notion from artificial intelligence. Moreover, it was shown in these papers that the presence of an $(l + 1)$-ary NU polymorphism f is equivalent to the l-Helly property for $\mathcal{B}$ and to definability of ¬CSP($\mathcal{B}$) by an (l, k)-Datalog program with a special property. Not going

into formal definitions here, the intuition behind these properties is that every relation invariant under f is decomposable into its l-fold projections, and so a mapping from $\mathcal{A}$ to $\mathcal{B}$ that is not a homomorphism can be shown to be such using only at most l-element subsets of $\mathcal{A}$.

Interestingly, up until now all known concrete structures $\mathcal{B}$ with CSP($\mathcal{B}$) in **NL** have an NU polymorphism. It is mentioned in [8] that all two-element structures $\mathcal{B}$ with an NU polymorphism (which precisely correspond to the first family mentioned in the introduction) have bounded path duality, and so the correponding problems CSP($\mathcal{B}$) belong to **NL**. All known concrete structures $\mathcal{B}$ that have at least three elements and such that CSP($\mathcal{B}$) in **NL** have a ternary NU polymorphism derived from dual discriminators (see Example 4) [8], these are the problems from the second family.

Our next result shows that every order constraint satisfaction problem with a near-unanimity polymorphism is in **NL**. Note that by [10], for every $l \geq 3$, there exists a poset that has an l-ary NU polymorphism, but no such polymorphism of smaller arity. Moreover, it was proved in [28] that it can be decided in polynomial time whether a poset has an NU polymorphism, while in [12] this was shown to be true for any fixed arity l for general structures.

In the following we deal mostly with posets, and so we use symbol $\leq$ (rather than R) to denote partial order. Recall that a poset is called *connected* if its comparability graph is connected.

An important tool used in the proof of the next theorem is the notion of a poset *zigzag* introduced in [45]. Intuitively, a $\mathcal{P}$-zigzag is a minimal obstruction for the extendibility problem Ext($\mathcal{P}$). More formally, a poset $\mathcal{X} = (X, \leq^{\mathcal{X}})$ is said to be contained in a poset $\mathcal{Q} = (Q, \leq^{\mathcal{Q}})$ if $\leq^{\mathcal{X}} \subseteq \leq^{\mathcal{Q}}$, and it is said to be properly contained if $\mathcal{X} \neq \mathcal{Q}$. A $\mathcal{P}$-*zigzag* is a pair $(\mathcal{Q}, f)$ such that f is a partial mapping from $\mathcal{Q}$ to $\mathcal{P}$ that cannot be extended to a full homomorphism, but, for every poset $\mathcal{X}$ properly contained in $\mathcal{Q}$, the mapping $f|_X$ is extendible to a full homomorphism from $\mathcal{X}$ to $\mathcal{P}$. The key fact in the proof of the following theorem is that a connected poset has a near-unanimity polymorphism if and only if the number of its zigzags is finite [30].

Theorem 4 *Let $\mathcal{P}$ be a poset with an NU polymorphism. Then* PoRet($\mathcal{P}$) *is in* **SL** *and* CSP($\mathcal{P}_P$) *is in* **NL**. *If, in addition, $\mathcal{P}$ is connected then* PoRet($\mathcal{P}$) *is in* **L**.

Theorem 4 generalizes a result from [38] where it was shown that if poset $\mathcal{P}$ has the 2-Helly property then PoRet($\mathcal{P}$) is in **NL**. Moreover, together with Theorem 2 it completely covers the classification of complexity of PoRet($\mathcal{P}$) for bipartite posets given in [38].

5   Series-Parallel Posets

In this section we exhibit the first concrete examples of structures without NU polymorphism but with constraint satisfaction problem solvable in non-deterministic logarithmic space. We deal with the poset-related problems for

series-parallel posets. These posets have been studied in computer science because they play an important role in concurrency (see, e.g., [32–34]).

Recall that a linear sum of two posets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a poset $\mathcal{P}_1 + \mathcal{P}_2$ with the universe $P_1 \cup P_2$ and partial order $\leq^{\mathcal{P}_1} \cup \leq^{\mathcal{P}_2} \cup \{(p_1, p_2) \mid p_1 \in P_1, p_2 \in P_2\}$.

Definition 4 *A poset is called* series-parallel *if it can be constructed from singletons by using disjoint union and linear sum.*

Let **k** denote a k-antichain (that is, disjoint union of k singletons). A 4-*crown* is a poset isomorphic to $\mathbf{2} + \mathbf{2}$. The N-*poset* can be described as $\mathbf{2} + \mathbf{2}$ with one comparability missing (its Hasse diagram looks like the letter "N"). Series-parallel posets can be characterized as N-free posets, that is, posets not containing the N-poset as a subposet [44].

Denote by $\mathbb{A}$ the class of all series-parallel posets $\mathcal{P}$ with the following property: if $\{a, a', b, b'\}$ is a 4-crown in $\mathcal{P}$, a and a' being the bottom elements, then at least one of the following conditions holds:

1. there is $e \in P$ such that a, a', e, b, b' form a subposet in $\mathcal{P}$ isomorphic to $\mathbf{2} + \mathbf{1} + \mathbf{2}$;
2. $\inf_{\mathcal{P}}(a, a')$ exists;
3. $\sup_{\mathcal{P}}(b, b')$ exists.

Recall that $\inf_{\mathcal{P}}$ and $\sup_{\mathcal{P}}$ denote the greatest common lower bound and the least common upper bound in $\mathcal{P}$, respectively.

Theorem 5 *Let $\mathcal{P}$ be a series-parallel poset. If $\mathcal{P} \in \mathbb{A}$ then* $\mathrm{PoRet}(\mathcal{P})$ *is in* **SL** *and* $\mathrm{CSP}(\mathcal{P}_P)$ *is in* **NL**. *Otherwise both problems are* **NP**-*complete (via polynomial-time reductions).*

The class $\mathbb{A}$ can be characterized by means of "forbidden retracts".

Lemma 1 *A series-parallel poset $\mathcal{P}$ belongs to $\mathbb{A}$ if and only if it has no retraction onto one of* $\mathbf{1} + \mathbf{2} + \mathbf{2} + \mathbf{2}$, $\mathbf{2} + \mathbf{2} + \mathbf{2} + \mathbf{1}$, $\mathbf{1} + \mathbf{2} + \mathbf{2} + \mathbf{2} + \mathbf{1}$, $\mathbf{2} + \mathbf{2}$, *and* $\mathbf{2} + \mathbf{2} + \mathbf{2}$,.

It was proved in [46] that a series-parallel poset $\mathcal{P}$ has an NU polymorphism if and only if it has no retraction onto one of $\mathbf{2} + \mathbf{2}$, $\mathbf{1} + \mathbf{2} + \mathbf{2}$, $\mathbf{2} + \mathbf{2} + \mathbf{1}$, and $\mathbf{1} + \mathbf{2} + \mathbf{2} + \mathbf{2} + \mathbf{1}$. Combining this result with Theorem 5 and Lemma 1 we can get, in particular, the following.

Corollary 2 *If $\mathcal{P}$ is one of* $\mathbf{1} + \mathbf{2} + \mathbf{2}$, $\mathbf{2} + \mathbf{2} + \mathbf{1}$, *and* $\mathbf{1} + \mathbf{2} + \mathbf{2} + \mathbf{2} + \mathbf{1}$ *then* $\mathrm{PoRet}(\mathcal{P})$ *is in* **SL** *and* $\mathrm{CSP}(\mathcal{P}_P)$ *is in* **NL** *but $\mathcal{P}_P$ has no NU polymorphism.*

It should be noted here that TC-feasibility of posets $\mathbf{2} + \mathbf{2} + \mathbf{1}$ and $\mathbf{1} + \mathbf{2} + \mathbf{2}$ was proved in [38].

In fact, we can say more about the complexity of $\mathrm{PoRet}(\mathcal{P})$ for the posets mentioned in Theorem 5.

Proposition 1 *If $\mathcal{P} \in \mathbb{A}$ and $\mathcal{P}$ has no NU polymorphisms then* $\mathrm{PoRet}(\mathcal{P})$ *is* **SL**-*complete (under many-one logarithmic space reductions).*

6 Conclusion

It is an open problem whether the presence of an NU polymorphism leads to solvabilty of a CSP($\mathcal{B}$) in non-deterministic logarithmic space for general structures. In this paper we solved the problem positively in the case of order constraints and also proved that this algebraic condition is not necessary. Hence, if the general problem above has a positive answer, then the algebraic condition characterizing CSPs in **NL** (if it exists) is of a more general form than an NU polymorphism.

Motivated by the problem, Dalmau asked [8] whether structures with an NU polymorphism have bounded path duality. At present, we do not know whether the structures $\mathcal{P}_P$ from the two classes described in this paper have bounded path duality. It follows from results of [9, 29] that if $\mathcal{P}_P$ has an NU polymorphism then $\neg \text{CSP}(\mathcal{P}_P)$ is definable in $(1, 2)$-Datalog, and the same can be shown for structures $\mathcal{P}_P$ with $\mathcal{P} \in \mathbb{A}$. There may be some way of using special properties of posets for transforming such $(1, 2)$-Datalog programs into linear Datalog programs.

It is easy to see that the posets from our two classes are TC-feasible (see definition in Section 3). It was shown in [38] that the class of TC-feasible posets is closed under isomorphism, dual posets, disjoint union, direct products, and retractions. Using these operations one can easily construct TC-feasible posets lying outside the two classes. For example, direct product of $\mathbf{2 + 2 + 1}$ and its dual poset neither has an NU polymorphism nor is series-parallel. In this way one can find further order constraint satisfaction problems in **NL**. Theorem 1 provides another way of obtaining further CSPs in **NL** because, clearly, the equality relation $=_P$ can be expressed in $\mathcal{P}_P$.

References

1. M. Alvarez and R. Greenlaw. A compendium of problems complete for symmetric logarithmic space. *Computational Complexity*, 9(2):123–145, 2000.
2. M. Benke. Some complexity bounds for subtype inequalities. *Theoretical Computer Science*, 212:3–27, 1999.
3. A.A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings 43rd IEEE Symposium on Foundations of Computer Science, FOCS'02*, pages 649–658, 2002.
4. A.A. Bulatov, A.A. Krokhin, and P.G. Jeavons. Constraint satisfaction problems and finite algebras. In *Proceedings 27th International Colloquium on Automata, Languages and Programming, ICALP'00*, volume 1853 of *Lecture Notes in Computer Science*, pages 272–282. Springer-Verlag, 2000.
5. A.A. Bulatov, A.A. Krokhin, and P.G. Jeavons. The complexity of maximal constraint languages. In *Proceedings 33rd ACM Symposium on Theory of Computing, STOC'01*, pages 667–674, 2001.

6. E. Corominas. Sur les ensembles ordonnés projectif et la propriétré du point fixe. *C. R. Acad. Sci. Paris Serie I Math.*, 311:199–204, 1990.

7. N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. 2001.

8. V. Dalmau. Constraint satisfaction problems in non-deterministic logarithmic space. In *Proceedings 29th International Colloquium on Automata, Languages and Programming, ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*, pages 414–425. Springer-Verlag, 2002.

9. V. Dalmau and J. Pearson. Set functions and width 1 problems. In *Proceedings 5th International Conference on Constraint Programming, CP'99*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer-Verlag, 1999.

10. J. Demetrovics, L. Hannák, and L. Rónyai. Near-unanimity functions of partial orders. In *Proceedings 14th International Symposium on Multiple-Valued Logic, ISMVL'84*, pages 52–56, 1984.

11. D. Duffus and I. Rival. A structure theory for ordered sets. *Discrete Mathematics*, 35:53–118, 1981.

12. T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal of Computing*, 28:57–104, 1998.

13. G. Gottlob, L. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124:243–282, 2000.

14. G. Gottlob, L. Leone, and F. Scarcello. Hypertree decomposition and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, 2002.

15. E. Grädel. Capturing complexity classes by fragments of second-order logic. *Theoretical Computer Science*, 101:35–57, 1992.

16. M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings 33rd ACM Symposium on Theory of Computing, STOC'01*, pages 657–666, 2001.

17. P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Ser.B*, 48:92–110, 1990.

18. M. Hoang and J.C. Mitchell. Lower bounds on type inference with subtypes. In *Proceedings 22nd ACM Symposium on Principles of Programming Languages, POPL'95*, pages 176–185, 1995.

19. N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17:935–939, 1988.

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

21. P.G. Jeavons, D.A. Cohen, and M.C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

22. P.G. Jeavons, D.A. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings 1st International Conference on Constraint Programming, CP'95*, volume 976 of *Lecture Notes in Computer Science*, pages 276–291. Springer-Verlag, 1995.

23. P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

24. L. Kirousis. Fast parallel constraint satisfaction. *Artificial Intelligence*, 64:147–160, 1993.

25. J. Köbler and J. Torán. The complexity of graph isomorphism for colored graphs with color classes of size 2 and 3. In *Proceedings 19th Symposium on Theoretical Aspects of Computer Science, STACS'02*, pages 121–132, 2002.

26. Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

27. V. Kumar. Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, 13(1):32–44, 1992.

28. G. Kun and Cs. Szabó. Order varieties and monotone retractions of finite posets. *Order*, 18:79–88, 2001.

29. B. Larose and L. Zádori. The complexity of the extendibility problem for finite posets. manuscript, obtainable from http://cicma.mathstat.concordia.ca/faculty/larose/.

30. B. Larose and L. Zádori. Algebraic properties and dismantlability of finite posets. *Discrete Mathematics*, 163:89–99, 1997.

31. H. Lewis and C. Papadimitriou. Symmetric space bounded computation. *Theoretical Computer Science*, 19:161–188, 1982.

32. K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages. In *Proceedings 15th Symposium on Theoretical Aspects of Computer Science, STACS'98*, pages 555–565, 1998.

33. R.H. Möhring. Computationally tractable classes of ordered sets. In *Algorithms and Order (Ottawa, 1987)*, pages 105–193. Kluwer, Dordrecht, 1989.

34. R.H. Möhring and M.W. Schäffter. Scheduling series-parallel orders subject to 0/1-communication delays. *Parallel Computing*, 25:23–40, 1999.

35. U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

36. N. Nisan and A. Ta-Shma. Symmetric logspace is closed under complementation. *Chicago Journal on Theoretical Computer Science*, 1, 1995. (electronic).

37. C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

38. V. Pratt and J. Tiuryn. Satisfiabilty of inequalities in a poset. *Fundamenta Informaticae*, 28:165–182, 1996.

39. R.W. Quackenbush, I. Rival, and I.G. Rosenberg. Clones, order varieties, near-unaminity functions and holes. *Order*, 7:239–248, 1990.

40. J.H. Reif. Symmetric complementation. In *Proceedings 14th ACM Symposium on Theory of Computing, STOC'82*, pages 201–214, 1982.

41. T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing, STOC'78*, pages 216–226, 1978.

42. E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, London, 1993.

43. J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1 & 2. Computer Science Press, 1989.

44. J. Valdes, R.E. Tarjan, and E.L. Lawler. The recognition of series-parallel digraphs. *SIAM Journal on Computing*, 11:298–313, 1982.

45. L. Zádori. Posets, near-unanimity functions and zigzags. *Bulletin of Australian Mathematical Society*, 47:79–93, 1993.

46. L. Zádori. Series parallel posets with nonfinitely generated clones. *Order*, 10:305–316, 1993.