Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 231–242
www.stacs-conf.org

ENUMERATING HOMOMORPHISMS

ANDREI A. BULATOV [1] AND VÍCTOR DALMAU [2] AND MARTIN GROHE [3] AND DÁNIEL MARX [4]

School of Computing Science, Simon Fraser University, Burnaby, Canada
E-mail address: abulatov@cs.sfu.ca

Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
E-mail address: victor.dalmau@tecn.upf.es

Institut für Informatik, Humboldt-Universität, Berlin, Germany
E-mail address: grohe@informatik.hu-berlin.de

Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Budapest, Hungary
E-mail address: dmarx@cs.bme.hu

ABSTRACT. The homomorphism problem for relational structures is an abstract way of formulating constraint satisfaction problems (CSP) and various problems in database theory. The decision version of the homomorphism problem received a lot of attention in literature; in particular, the way the graph-theoretical structure of the variables and constraints influences the complexity of the problem is intensively studied. Here we study the problem of enumerating all the solutions with polynomial delay from a similar point of view. It turns out that the enumeration problem behaves very differently from the decision version. We give evidence that it is unlikely that a characterization result similar to the decision version can be obtained. Nevertheless, we show nontrivial cases where enumeration can be done with polynomial delay.

1. Introduction

Constraint satisfaction problems (CSP) form a rich class of algorithmic problems with applications in many areas of computer science. We only mention database systems, where CSPs appear in the guise of the conjunctive query containment problem and the closely related problem of evaluating conjunctive queries. It has been observed by Feder and Vardi [14] that as abstract problems, CSPs are homomorphism problems for relational structures. Algorithms for and the complexity of constraint satisfaction problems have been intensely studied (e.g. [20, 10, 4, 5]), not only for the standard decision problems but also optimization versions (e.g. [3, 22, 23, 24]) and counting versions (e.g. [6, 7, 8, 13]) of CSPs.

In this paper we study the *CSP enumeration problem*, that is, problem of computing all solutions for a given CSP instance. More specifically, we are interested in the question which structural restrictions on CSP instances guarantee tractable enumeration problems. "Structural restrictions"

The second author is supported by the MCyT through grants TIN2006-15387-C03-03 and TIN2007-68005-C04-03, and the program José Castillejo. Research of the fourth author is supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (Grant Number OTKA 67651).

STACS 2009
Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 231-242
http://drops.dagstuhl.de/opus/volltexte/2009/1838

are restrictions on the structure induced by the constraints on the variables. Example of structural restrictions is "every variable occurs in at most 5 constraints" or "the constraints form an acyclic hypergraph.[1]" This can most easily be made precise if we view CSPs as homomorphism problems: Given two relational structures $\mathbb{A}, \mathbb{B}$, decide if there is a homomorphism from $\mathbb{A}$ to $\mathbb{B}$. Here the elements of the structure $\mathbb{A}$ correspond to the variables of the CSP and the elements of the structure $\mathbb{B}$ correspond to the possible values. Structural restrictions are restrictions on the structure $\mathbb{A}$. If $\mathcal{A}$ is a class of structures, then $\mathrm{CSP}(\mathcal{A}, -)$ denotes the restriction of the general CSP (or homomorphism problem) where the "left hand side" input structure $\mathbb{A}$ is taken from the class $\mathcal{A}$. $\mathrm{ECSP}(\mathcal{A}, -)$ denotes the corresponding enumeration problem: Given two relational structures $\mathbb{A} \in \mathcal{A}$ and $\mathbb{B}$, compute the set of all homomorphisms from $\mathbb{A}$ to $\mathbb{B}$. The enumeration problem is of particular interest in the database context, where we are usually not only interested in the question of whether the answer to a query is nonempty, but want to compute all tuples in the answer. We will also briefly discuss the corresponding *search* problem: Find a solution if one exists, denoted $\mathrm{SCSP}(\mathcal{A}, -)$.

It has been shown in [2] that $\mathrm{ECSP}(\mathcal{A}, -)$ can be solved in polynomial time if and only if the number of solutions (that is, homomorphisms) for all instances is polynomially bounded in terms of the input size and that this is the case if and only if the structures in the class $\mathcal{A}$ have bounded fractional edge cover number. However, usually we cannot expect the number of solutions to be polynomial. In this case, we may ask which conditions on $\mathcal{A}$ guarantee that $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm. A *polynomial delay algorithm* for an enumeration problem is required to produce the first solution in polynomial time and then iteratively compute all solutions (each solution only once), leaving only polynomial time between two successive solutions. In particular, this guarantees that the algorithms computes all solutions in *polynomial total time*, that is, in time polynomial in the input size plus output size.

It is easy to see that $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm if the class $\mathcal{A}$ has bounded tree width. It is also easy to see that there are classes $\mathcal{A}$ of unbounded tree width such that $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm. It follows from our results that examples of such classes are the class of all grids or the class of all complete graphs with a loop on every vertex. It is known that the decision problem $\mathrm{CSP}(\mathcal{A}, -)$ is in polynomial time if and only if the cores of the structures in $\mathcal{A}$ have bounded tree width [17] (provided the arity of the constraints is bounded, and under some reasonable complexity theoretic assumptions). A *core* of a relational structure $\mathcal{A}$ is a minimal substructure $\mathcal{A}' \subseteq \mathcal{A}$ such that there is a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$; minimality is with respect to inclusion. It is easy to see that all cores of a structure are isomorphic. Hence we usually speak of "the" core of a structure. Note that the core of a grid (and of any other bipartite graph with at least one edge) is a single edge, and the core of a complete graph with all loops present (and of any other graph with a loop) is a single vertex with a loop on it. The core of a complete graph with no loops is the graph itself. As a polynomial delay algorithm for an enumeration algorithms yields a polynomial time algorithm for the corresponding decision problem, it follows that $\mathrm{ECSP}(\mathcal{A}, -)$ can only have a polynomial delay algorithm if the cores of the structures in $\mathcal{A}$ have bounded tree width. Unfortunately, there are examples of classes $\mathcal{A}$ that have cores of bounded tree width, but for which $\mathrm{ECSP}(\mathcal{A}, -)$ has no polynomial delay algorithm unless P $=$ NP (see Example 3.2).

Our main algorithmic results show that $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm if the cores of the structures in $\mathcal{A}$ have bounded tree width and if, in addition, they can be reached in a sequence of "small steps." An *endomorphism* of a structure is a homomorphism of a structure to itself. A *retraction* is an endomorphism that is the identity mapping on its image. Every structure

[1]The other type of restrictions studied in the literature on CSP are "constraint language restrictions", that is, restrictions on the structure imposed by the constraint relations on the values. An example of a constraint language restriction is "all clauses of a SAT instance, viewed as a Boolean CSP, are Horn clauses".

has a retraction to its core. However, in general, the only way to map a structure to its core may be by collapsing the whole structure at once. As an example, consider a path with a loop on both endpoints. The core consists of a single vertex with a loop. (More precisely, the two cores are the two endpoints with their loops.) The only endomorphism of this structure to a proper substructure maps the whole structure to its core. Compare this with a path that only has a loop on one endpoint. Again, the core is a single vertex with a loop, but now we can reach the core by a sequence of retractions, mapping a path of length n to a subpath of length $n-1$ and then to a subpath of length $n-2$ et cetera. We prove that if $\mathcal{A}$ is a class of structures whose cores have bounded tree width and can be reached by a sequence of retractions each of which only moves a bounded number of vertices, then $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm.

We also consider more general sequences of retractions or endomorphism from a structure to its core. We say that a sequence of endomorphisms from a structure $\mathbb{A}_0$ to a substructure $\mathbb{A}_1 \subset \mathbb{A}_0$, from $\mathbb{A}_1$ to a substructure $\mathbb{A}_2, \ldots,$ to a structure $\mathbb{A}_n$ has *bounded width* if $\mathbb{A}_n$ and, for each $i \leq n$, the "difference between $\mathbb{A}_i$ and $\mathbb{A}_{i-1}$" has bounded tree width. We prove that if we are given a sequence of endomorphisms of bounded width together with the input structure $\mathbb{A}$, then we can compute all solutions by a polynomial delay algorithm. Unfortunately, in general we cannot compute such a sequence of endomorphisms efficiently. We prove that even for width 1 it is NP-complete to decide whether such a sequence exists.

Finally, we remark that our results are far from giving a complete classification of the classes $\mathcal{A}$ for which $\mathrm{ECSP}(\mathcal{A}, -)$ has a polynomial delay algorithm and those classes for which it does not. Indeed, we show that it will be difficult to obtain such a classification, because such a classification would imply a solution to the notoriously open *CSP dichotomy conjecture* of Feder and Vardi [14] (see Section 3 for details).

Due to space restrictions several proofs are omitted.

2. Preliminaries

Relational structures. A *vocabulary* τ is a finite set of *relation symbols* of specified arities. A *relational structure* $\mathbb{A}$ over τ consists of a finite set A called the *universe* of $\mathbb{A}$ and for each relation symbol $R \in \tau$, say, of arity r, an r-ary relation $R^{\mathbb{A}} \subseteq A^r$. Note that we require vocabularies and structures to be finite. A structure $\mathbb{A}$ is a *substructure* of a structure $\mathbb{B}$ if $A \subseteq B$ and $R^{\mathbb{A}} \subseteq R^{\mathbb{B}}$ for all $R \in \tau$. We write $\mathbb{A} \subseteq \mathbb{B}$ to denote that $\mathbb{A}$ is a substructure of $\mathbb{B}$ and $\mathbb{A} \subset \mathbb{B}$ to denote that $\mathbb{A}$ is a *proper* substructure of $\mathbb{B}$, that is, $\mathbb{A} \subseteq \mathbb{B}$ and $\mathbb{A} \neq \mathbb{B}$. A substructure $\mathbb{A} \subseteq \mathbb{B}$ is *induced* if for all $R \in \tau$, say, of arity r, we have $R^{\mathbb{A}} = R^{\mathbb{B}} \cap A^r$. For a subset $A \subseteq B$, we write $\mathbb{B}[A]$ to denote the induced substructure of $\mathbb{B}$ with universe A.

Homomorphisms. We often abbreviate tuples $(a_1, \ldots, a_k)$ by $\mathbf{a}$. If f is a mapping whose domain contains $a_1, \ldots, a_k$ we write $f(\mathbf{a})$ to abbreviate $(f(a_1), \ldots, f(a_k))$. A *homomorphism* from a relational structure $\mathbb{A}$ to a relational structure $\mathbb{B}$ is a mapping $\varphi : A \to B$ such that for all $R \in \tau$ and all tuples $\mathbf{a} \in R^{\mathbb{A}}$ we have $\varphi(\mathbf{a}) \in R^{\mathbb{B}}$. A *partial homomorphism* on $C \subseteq A$ to $\mathbb{B}$ is a homomorphism of $\mathbb{A}[C]$ to $\mathbb{B}$. It is sometimes useful when designing examples to exclude certain homomorphisms or endomorphisms. The simplest way to do that is to use unary relations. For example, if R is a unary relation and $(a) \in R^{\mathbb{A}}$ we say that a *has color* R. Now if $b \in B$ does not have color R then no homomorphism from $\mathbb{A}$ to $\mathbb{B}$ maps a to b.

Two structures $\mathbb{A}$ and $\mathbb{B}$ are *homomorphically equivalent* if there is a homomorphism from $\mathbb{A}$ to $\mathbb{B}$ and also a homomorphism from $\mathbb{B}$ to $\mathbb{A}$. Note that if structures $\mathbb{A}$ and $\mathbb{A}'$ are homomorphically equivalent, then for every structure $\mathbb{B}$ there is a homomorphism from $\mathbb{A}$ to $\mathbb{B}$ if and only if there is a homomorphism from $\mathbb{A}'$ to $\mathbb{B}$; in other words: the instances $(\mathbb{A}, \mathbb{B})$ and $(\mathbb{A}', \mathbb{B})$ of the decision CSP are equivalent. However, the two instances may have vastly different sizes, and the complexity of solving the search and enumeration problems for them can also be quite different. Homomorphic equivalence is closely related to the concept of the core of a structure: A structure $\mathbb{A}$ is a *core* if there is no homomorphism from $\mathbb{A}$ to a proper substructure of $\mathbb{A}$. A core of a structure $\mathbb{A}$ is a substructure $\mathbb{A}' \subseteq \mathbb{A}$ such that there is a homomorphism from $\mathbb{A}$ to $\mathbb{A}'$ and $\mathbb{A}'$ is a core. Obviously, every core of a structure is homomorphically equivalent to the structure. We observe another basic fact about cores:

Observation 2.1. Let $\mathbb{A}$ and $\mathbb{B}$ be homomorphically equivalent structures, and let $\mathbb{A}'$ and $\mathbb{B}'$ be cores of $\mathbb{A}$ and $\mathbb{B}$, respectively. Then $\mathbb{A}'$ and $\mathbb{B}'$ are isomorphic. In particular, all cores of a structure $\mathbb{A}$ are isomorphic. Therefore, we often speak of *the* core of $\mathbb{A}$.

Observation 2.2. It is easy to see that it is NP-hard to decide, given structures $\mathbb{A} \subseteq \mathbb{B}$, whether $\mathbb{A}$ is isomorphic to the core of $\mathbb{B}$. (For an arbitrary graph G, let $\mathbb{A}$ be a triangle and $\mathbb{B}$ the disjoint union of G with $\mathbb{A}$. Then $\mathbb{A}$ is a core of $\mathbb{B}$ if and only if G is 3-colorable.) Hell and Nešetřil [19] proved that it is co-NP-complete to decide whether a graph is a core.

Tree decompositions. A *tree decomposition* of a graph G is a pair (T, B), where T is a tree and B is a mapping that associates with every node $t \in V(T)$ a set $B_t \subseteq V(G)$ such that (1) for every $v \in V(G)$ the set $\{t \in V(T) | v \in B_t\}$ is connected in T, and (2) for every $e \in E(G)$ there is a $t \in V(T)$ such that $e \subseteq B_t$. The sets B_t, for $t \in V(T)$, are called the *bags* of the decomposition. It is sometimes convenient to have the tree T in a tree decomposition rooted; we always assume it is. The *width* of a tree decomposition (T, B) is $\max\{|B_t| \mid t \in V(T)\} - 1$. The *tree width* of a graph G, denoted by tw(G), is the minimum of the widths of all tree decompositions of G.

We need to transfer some of the notions of graph theory to arbitrary relational structures. The *Gaifman graph* (also known as *primal graph*) of a relational structure $\mathbb{A}$ with vocabulary τ is the graph $G(\mathbb{A})$ with vertex set $\mathbb{A}$ and an edge between a and b if $a \neq b$ and there is a relation symbol $R \in \tau$, say, of arity r, and a tuple $(a_1, \ldots, a_r) \in R^{\mathbb{A}}$ such that $a, b \in \{a_1, \ldots, a_r\}$. We can now transfer graph-theoretic notions to relational structures. In particular, a subset $B \subseteq A$ is *connected* in a structure $\mathbb{A}$ if it is connected in $G(\mathbb{A})$. A *tree decomposition* of a structure $\mathbb{A}$ can simply be defined to be a tree-decomposition of $G(\mathbb{A})$. Equivalently, a tree decomposition of $\mathbb{A}$ can be defined directly by replacing the second condition in the definition of tree decompositions of graphs by (2') for every $R \in \tau$ and $(a_1, \ldots, a_r) \in R^{\mathbb{A}}$ there is a $t \in V(T)$ such that $\{a_1, \ldots, a_r\} \subseteq B_t$. A class $\mathcal{C}$ of structures has *bounded tree width* if there is a $w \in \mathbb{N}$ such that tw$(\mathbb{A}) \leq w$ for all $\mathbb{A} \in \mathcal{C}$. A class $\mathcal{C}$ of structures has *bounded tree width modulo homomorphic equivalence* if there is a $w \in \mathbb{N}$ such that every $\mathbb{A} \in \mathcal{C}$ is homomorphically equivalent to a structure of tree width at most w.

Observation 2.3. A structure $\mathbb{A}$ is homomorphically equivalent to a structure of tree width at most w if and only if the core of $\mathbb{A}$ has tree width at most w.

The Constraint Satisfaction Problem. For two classes $\mathcal{A}$ and $\mathcal{B}$ of structures, the *Constraint Satisfaction Problem*, $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$, is the following problem:

$\mathrm{CSP}(\mathcal{A}, \mathcal{B})$
Instance: $\mathbb{A} \in \mathcal{A}, \mathbb{B} \in \mathcal{B}$
Problem: Decide if there is a homomorphism from $\mathbb{A}$ to $\mathbb{B}$.

The CSP is a decision problem. The variation of it we study in this paper is the following enumeration problem:

$\mathrm{ECSP}(\mathcal{A},\mathcal{B})$
Instance: $\mathbb{A}\in\mathcal{A}, \mathbb{B}\in\mathcal{B}$
Problem: Output all the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$.

We shall also refer to the search problem, $\mathrm{SCSP}(\mathcal{A},\mathcal{B})$, in which the goal is to find one solution to a CSP-instance or output 'no' if a solution does not exists.

If one of the classes $\mathcal{A}$, $\mathcal{B}$ is the class of all finite structures, then we denote the corresponding CSPs by $\mathrm{CSP}(\mathcal{A},-)$, $\mathrm{CSP}(-,\mathcal{B})$ (respectively, $\mathrm{ECSP}(\mathcal{A},-)$, $\mathrm{ECSP}(-,\mathcal{B})$, $\mathrm{SCSP}(\mathcal{A},-)$, $\mathrm{SCSP}(-,\mathcal{B})$).

The decision CSP has been intensely studied. If a class $\mathcal{C}$ of structures has bounded arity then $\mathrm{CSP}(\mathcal{C},-)$ is solvable in polynomial time if and only if $\mathcal{C}$ has bounded tree width modulo homomorphic equivalence [17]. If the arity of $\mathcal{C}$ is not bounded, several quite general conditions on a class of structures have been identified that guarantee polynomial time solvability of $\mathrm{CSP}(\mathcal{C},-)$, see, e.g.[16, 12, 18]. Problems of the form $\mathrm{CSP}(-,\mathcal{C})$ have been studied mostly in the case when $\mathcal{C}$ is 1-element. Problems of this type are sometimes referred to as *non-uniform*. It is conjectured that every non-uniform problem is either solvable in polynomial time or NP-complete (the so-called *Dichotomy Conjecture*) [14]. Although this conjecture is proved in several particular cases [20, 9, 10, 4], in its general form it is believed to be very difficult.

A search CSP is clearly no easier than the corresponding decision problem. While any non-uniform search problem $\mathrm{SCSP}(-,\mathcal{C})$ is polynomial time reducible to its decision version $\mathrm{CSP}(-,\mathcal{C})$ [11], nothing is known about the complexity of search problems $\mathrm{SCSP}(\mathcal{C},-)$ except the result we state in Section 3. Paper [25] provides some initial results on the complexity of non-uniform enumerating problems.

3. Tractable structures for enumeration

Since even an easy CSP may have exponentially many solutions, the model of choice for 'easy' enumeration problems is algorithms with polynomial delay [21]. An algorithm Alg is said to solve a CSP *with polynomial delay* (*WPD* for short) if there is a polynomial $p(n)$ such that, for every instance of size n, Alg outputs 'no' in a time bounded by $p(n)$ if there is no solution, otherwise it generates all solutions to the instance such that no solution is output twice, the first solution is output after at most $p(n)$ steps after the computation starts, and time between outputting two consequent solutions does not exceed $p(n)$.

If a class of relational structures $\mathcal{C}$ has bounded arity, the aforementioned result of Grohe [17] imposes strong restrictions on enumeration problems solvable WPD.

Observation 3.1. If a class of relational structures $\mathcal{C}$ with bounded arity does not have bounded tree width modulo homomorphic equivalence, then $\mathrm{ECSP}(\mathcal{C},-)$ is not WPD, unless P=NP.

Unlike for the decision version, the converse is not true: bounded tree width modulo homomorphic equivalence does not imply enumerability WPD.

Example 3.2. Let $\mathbb{A}_k$ be the disjoint union of a k-clique and a loop and let $\mathcal{A}=\{\mathbb{A}_k \mid k\geq 1\}$. Clearly, the core of each graph in $\mathcal{A}$ has bounded tree width (in fact, it is a single element), hence $\mathrm{CSP}(\mathcal{A},-)$ is polynomial-time solvable. For an arbitrary graph $\mathbb{B}$ without loops, let $\mathbb{B}'$ be the disjoint union of $\mathbb{B}$ and a loop. It is clear that there is always a trivial homomorphism from $\mathbb{A}_k$ (for any $k \geq 1$) to $\mathbb{B}'$ that maps everything into the loop. There exist homomorphisms different from the trivial one if and only if $\mathbb{B}$ contains a k-clique. Thus if we are able to check in polynomial time whether there is a second homomorphism, then we are able to test if $\mathbb{B}$ has a k-clique. Therefore, although $\mathrm{CSP}(\mathcal{A}, -)$ and $\mathrm{SCSP}(\mathcal{A}, -)$ are polynomial-time solvable, a WPD enumeration algorithm for $\mathrm{ECSP}(\mathcal{A}, -)$ would imply P $=$ NP.

It is not difficult to show that $\mathrm{ECSP}(\mathcal{C}, -)$ is enumerable WPD if $\mathcal{C}$ has bounded tree width. For space restrictions we do not include a direct proof and instead we derive it from a more general result in Section 4. Thus enumerability WPD has a different tractability criterion than the decision version, and this criterion lies somewhere between bounded tree width and bounded tree width modulo homomorphic equivalence. Thus in order to ensure that the solutions can be enumerated WPD, we have to make further restrictions on the way the structure can be mapped to its bounded tree width core. The main new definition of the paper requires that the core is reached by "small steps":

Let $\mathbb{A}$ be a relational structure with universe A. We say that $\mathbb{A}$ *has a sequence of endomorphisms of width* k if there are subsets $A = A_0 \supset A_1 \supset \ldots \supset A_n \neq \emptyset$ and homomorphisms $\varphi_1, \ldots, \varphi_n$ such that

(1) φ_i is a homomorphism from $\mathbb{A}[A_{i-1}]$ to $\mathbb{A}[A_i]$,
(2) $\varphi_i(A_{i-1}) = A_i$ for $1 \leq i \leq n$;
(3) if G is the primal graph of $\mathbb{A}$, then the tree width of $G[A_i \setminus A_{i+1}]$ is at most k for every $0 \leq i < n$;
(4) the structure induced by A_n has tree width at most k.

In Section 4, we show that enumeration for $(\mathbb{A}, \mathbb{B})$ can be done WPD if a sequence of bounded width endomorphisms for $\mathbb{A}$ is given in the input. Unfortunately, we cannot claim that $\mathrm{ECSP}(\mathcal{A}, -)$ can be done WPD if every structure in $\mathcal{A}$ has such a sequence, since we do not know how to find such sequences efficiently. In fact, as we show in Section 5, it is hard to check if a width-1 sequence exists for a given structure. Furthermore, we show a class $\mathcal{A}$ where every structure has a width-2 sequence, but $\mathrm{ECSP}(\mathcal{A}, -)$ cannot be done WPD, unless P $=$ NP. This means that it is not possible to get around the problem of not being able to find the sequences (for example, by finding sequences with somewhat larger width or by constructing the sequence during the enumeration).

Thus having a bounded width sequence of endomorphisms is not the right tractability criterion. We then investigate a more restrictive notion, where the bound is not on the tree width of the difference of the layers but on the number of elements in the differences. However, in the rest of the section, we give evidence that enumeration problems solvable WPD cannot be characterized in simple terms relying on tree width. For instance, a description of search problems solvable in polynomial time would imply a description of non-uniform decision problems solvable in polynomial time. This is shown via an analogous result for the search version of the problem, which might be of independent interest. By $\mathbb{A} \oplus \mathbb{B}$ we denote the disjoint union of relational structures $\mathbb{A}$ and $\mathbb{B}$.

Lemma 3.3. *Let $\mathbb{B}$ be a relational structure, which is a core, and let $\mathcal{C}_{\mathbb{B}}$ be $\{\mathbb{A} \oplus \mathbb{B} \mid \mathbb{A} \to \mathbb{B}\}$. Then $\mathrm{CSP}(-, \mathbb{B})$ is solvable in polynomial time if and only if so is the problem $\mathrm{SCSP}(\mathcal{C}_{\mathbb{B}}, -)$.*

Proof. If the decision problem $\mathrm{CSP}(-, \mathbb{B})$ is solvable in polynomial time we can construct an algorithm that given an instance $(\mathbb{A}, \mathbb{C})$ of $\mathrm{CSP}(\mathcal{C}_{\mathbb{B}}, -)$ computes a solution in polynomial time. Indeed, as $\mathrm{CSP}(-, \mathbb{B})$ is solvable in polynomial time by the aforementioned result of [11] it is also polynomial time to find a homomorphism from a given structure to $\mathbb{B}$ provided one exists. If $\mathbb{A} \in \mathcal{C}_{\mathbb{B}}$ such a homomorphism φ exists by the definition of $\mathcal{C}_{\mathbb{B}}$. So our algorithms, first, finds some homomorphism φ. Then it decides by brute force whether or not there exists a homomorphism φ' from $\mathbb{B}$ to $\mathbb{C}$ (note that this can be done in polynomial time for every fixed $\mathbb{B}$). If such a homomorphism does not exist then we can certainly guarantee that there is no homomorphism from $\mathbb{A}$ to $\mathbb{C}$. Otherwise we obtain a required homomorphism ψ as follows: Let $\psi(a) = \varphi'(a)$ for $a \in \mathbb{B}$, and $\psi(a) = \varphi' \circ \varphi(a)$ for $a \in \mathbb{A}$.

Conversely, assume that we have an algorithm Alg that finds a solution of any instance of $\mathrm{CSP}(\mathcal{C}_{\mathbb{B}}, -)$ in polynomial time, say, $p(n)$. We construct from it an algorithm that solves $\mathrm{CSP}(-, \mathbb{B})$. Given an instance $(\mathbb{A}, \mathbb{B})$ of $\mathrm{CSP}(-, \mathbb{B})$ we call algorithm Alg with input $\mathbb{A} \oplus \mathbb{B}$ and $\mathbb{B}$. Additionally we count the number of steps performed by Alg in such a way that we stop if Alg has not finished in $p(n)$ steps. If Alg produces a correct answer then we have to be able to obtain from it a homomorphism from $\mathbb{A}$ to $\mathbb{B}$. If Alg's answer is not correct or the clock reaches $p(n)$ steps we know that Alg failed. The only possible reason for that is that $\mathbb{A} \oplus \mathbb{B}$ does not belong to $\mathcal{C}_{\mathbb{B}}$, which implies that $\mathbb{A}$ is not homomorphic to $\mathbb{B}$. ∎

In what follows we transfer this result to enumeration problems. Let $\mathcal{A}$ be a class of relational structures. The class $\mathcal{A}'$ consists of all structures built as follows: Take $\mathbb{A} \in \mathcal{A}$ and add to it $|\mathbb{A}|$ independent vertices.

Lemma 3.4. *Let $\mathcal{A}$ be a class of relational structures. Then $\mathrm{SCSP}(\mathcal{A}, -)$ is solvable in polynomial time if and only if $\mathrm{ECSP}(\mathcal{A}', -)$ is solvable WPD.*

Proof. If $\mathrm{ECSP}(\mathcal{A}, -)$ is enumerable WPD, then for any structure $\mathbb{A}' \in \mathcal{A}'$ it takes time polynomial in $|\mathbb{A}'|$ to find the first solution. Since $\mathbb{A}'$ is only twice of the size of the corresponding structure $\mathbb{A}$, it takes only polynomial time to solve $\mathrm{SCSP}(\mathcal{A}, -)$.

Conversely, given a structure $\mathbb{A}' = \mathbb{A} \cup I \in \mathcal{A}'$, where $\mathbb{A} \in \mathcal{A}$ and I is the set of independent elements, and any structure $\mathbb{B}$. The first homomorphism from $\mathbb{A}'$ to $\mathbb{B}$ can be found in polynomial time, since $\mathrm{SCSP}(\mathcal{A}, -)$ is polynomial time solvable and the independent vertices can be mapped arbitrarily. Let the restriction of this homomorphism onto $\mathbb{A}$ be φ. Then while enumerating all possible $|\mathbb{B}|^{|\mathbb{A}|}$ extensions of φ we buy enough time to enumerate all homomorphisms from $\mathbb{A}$ to $\mathbb{B}$ using brute force. ∎

4. Sequence of bounded width endomorphisms

In this section we show that for every fixed k, all the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$ can be enumerated with polynomial delay if a sequence of width k endomorphisms of $\mathbb{A}$ is given in the input. Given a sequence $A_0, \ldots, A_n$ and $\varphi_1, \ldots, \varphi_n$ as in the definition of a sequence of width k endomorphisms, we denote $\mathbb{A}[A_i]$ by $\mathbb{A}_i$.

We will enumerate the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$ by first enumerating the homomorphisms from $\mathbb{A}_n$, $\mathbb{A}_{n-1}$, $\ldots$ to $\mathbb{B}$ and then transforming them to homomorphisms from $\mathbb{A}$ to $\mathbb{B}$ using the homomorphisms φ_i. We obtain the homomorphisms from $\mathbb{A}_i$ by extending the homomorphism from $\mathbb{A}_{i+1}$ to the set $A_i \setminus A_{i+1}$; Lemma 4.1 below will be useful for this purpose. In order to avoid producing a homomorphism multiple times, we need a delicate classification (see definitions of elementary homomorphisms and of the index of a homomorphism).

Lemma 4.1. *Let $\mathbb{A}, \mathbb{B}$ be relational structures and $X_1 \subseteq X_2 \subseteq A$ subsets, and let g_0 be a homomorphism from $\mathbb{A}[X_1]$ to B. For every fixed k, there is a polynomial-time algorithm* HOMOMORPHISM-EXT$(\mathbb{A}, \mathbb{B}, X_1, X_2, g_0)$ *that decides whether g_0 can be extended to a homomorphism from $\mathbb{A}[X_2]$ to $\mathbb{B}$, if the tree width of induced subgraph $G[X_2 \setminus X_1]$ of the Gaifman graph of $\mathbb{A}$ is at most k.*

The *index* of a homomorphism φ from $\mathbb{A}$ to $\mathbb{B}$ is the largest t such that φ can be written as $\varphi = \psi \circ \varphi_t \circ \ldots \circ \varphi_1$ for some homomorphism ψ from $\mathbb{A}_t$ to $\mathbb{B}$. In particular, if φ cannot be written as $\varphi = \psi \circ \varphi_1$, then the index of φ is 0. Observe that if the index of φ is at least t, then there is a unique ψ such that $\varphi = \psi \circ \varphi_t \circ \ldots \circ \varphi_1$: This follows from the fact that $\varphi_t \circ \ldots \circ \varphi_1$ is a surjective mapping from A to A_t, thus if ψ' and ψ'' differ on A_t, then $\psi' \circ \varphi_t \circ \ldots \circ \varphi_1$ and $\psi'' \circ \varphi_t \circ \ldots \circ \varphi_1$ differ on A. A homomorphism ψ from $\mathbb{A}_t$ to $\mathbb{B}$ is *elementary*, if it cannot be written as $\psi = \psi' \circ \varphi_{t+1}$. A homomorphism is *reducible* if it is not elementary.

Lemma 4.2. *If a homomorphism ψ from $\mathbb{A}_t$ to $\mathbb{B}$ is elementary, then $\varphi = \psi \circ \varphi_t \circ \ldots \circ \varphi_1$ has index exactly t. Conversely, if homomorphism φ from $\mathbb{A}$ to $\mathbb{B}$ has index t and can be written as $\varphi = \psi \circ \varphi_t \circ \ldots \circ \varphi_1$, then the homomorphism ψ from $\mathbb{A}_t$ to $\mathbb{B}$ is elementary.*

Lemma 4.2 suggests a way of enumerating all the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$: for $t = 0, \ldots, n$, we enumerate all the elementary homomorphisms from $\mathbb{A}_t$ to $\mathbb{B}$, and for each such homomorphism ψ, we compute $\varphi = \psi \circ \varphi_t \circ \ldots \circ \varphi_1$. To this end, we need the following characterization of elementary homomorphisms:

Lemma 4.3. *A homomorphism ψ from $\mathbb{A}_t$ to $\mathbb{B}$ is reducible if and only if*

(1) *$\psi(x) = \psi(y)$ for every $x, y \in A_t$ with $\varphi_{t+1}(x) = \varphi_{t+1}(y)$, i.e., for every $z \in A_{t+1}$, $\psi(x)$ has the same value b_z for every x with $\varphi_{t+1}(x) = z$, and*
(2) *the mapping defined by $\psi'(z) := b_z$ is a homomorphism from $\mathbb{A}_{t+1}$ to $\mathbb{B}$.*

Lemma 4.3 gives a way of testing in polynomial time whether a given homomorphism ψ is elementary: we have to test whether one of the two conditions are violated. We state this in a more general form: we can test in polynomial time whether a partial mapping g_0 can be extended to an elementary homomorphism ψ, if the structure induced by the elements where g_0 is not defined has bounded tree width. We fix values every possible way in which the conditions of Lemma 4.3 can be violated and use HOMOMORPHISM-EXT to check whether there is an extension compatible with this choice. In order to efficiently enumerate all the possible violations of the second condition, the following definition is needed:

Given a relation $R^{\mathbb{B}}$ of arity r, a *bad prefix* is a tuple $(b_1, \ldots, b_s) \in B^s$ with $s \leq r$ such that

(1) there is no tuple $(b_1, \ldots, b_s, b_{s+1}, \ldots, b_r) \in R^{\mathbb{B}}$ for any $b_{s+1}, \ldots, b_r \in B$, and
(2) there is a tuple $(b_1, \ldots, b_{s-1}, c_s, c_{s+1}, \ldots, c_r) \in R^{\mathbb{B}}$ for some $c_t, \ldots, c_r \in B$.

If $(b_1, \ldots, b_r) \notin R^{\mathbb{B}}$, then there is a unique $1 \leq s \leq r$ such that the tuple $(b_1, \ldots, b_s)$ is a bad prefix: there has to be an s such that $(b_1, \ldots, b_s)$ cannot be extended to a tuple of $R^{\mathbb{B}}$, but $(b_1, \ldots, b_{s-1})$ can.

Lemma 4.4. *The relation $R^{\mathbb{B}}$ has at most $|R^{\mathbb{B}}| \cdot (|B| - 1) \cdot r$ bad prefixes, where r is the arity of the relation.*

Lemma 4.5. *Let X be a subset of A_t and let g_0 be a mapping from X to B. For every fixed k, there is a polynomial-time algorithm* ELEMENTARY-EXT(t, X, g_0) *that decides whether g_0 can be extended to an elementary homomorphism from $\mathbb{A}_t$ to B, if the tree width of the structure induced by $A_t - X$ is at most k.*

We enumerate the elementary homomorphisms in a specific order defined by the following precedence relation. Let φ be an elementary homomorphism from $\mathbb{A}_i$ to $\mathbb{B}$ and let ψ be an elementary homomorphism from $\mathbb{A}_j$ to $\mathbb{B}$ for some $j > i$. Homomorphism ψ is the *parent* of φ (φ is a *child* of ψ) if φ restricted to A_{i+1} can be written as $\psi \circ \varphi_j \circ \ldots \circ \varphi_{i+2}$. *Ancestor* and *descendant* relations are defined as the reflexive transitive closure of the parent and child relations, respectively.

Note that an elementary homomorphism from $\mathbb{A}_i$ to $\mathbb{B}$ has exactly one parent for $i < n$ and a homomorphism from $\mathbb{A}_n$ to $\mathbb{B}$ has no parent. Fix an arbitrary ordering of the elements of A. For $0 \leq i \leq n$ and $0 \leq j \leq |A_i \setminus A_{i+1}|$, let $A_{i,j}$ be the union of A_{i+1} and the first j elements of $A_i \setminus A_{i+1}$. Note that $A_{i,0} = A_{i+1}$ and $A_{i,|A_i \setminus A_{i+1}|} = A_i$.

Lemma 4.6. *Let ψ be a mapping from $A_{i,j}$ to $\mathbb{B}$ that can be extended to an elementary homomorphism from $\mathbb{A}_i$ to $\mathbb{B}$. Assume that a sequence of width k endomorphisms is given for $\mathbb{A}$. For every fixed k, there is a polynomial-delay, polynomial-space algorithm* ELEMENTARY-ENUM(i, j, ψ) *that enumerates all the elementary homomorphisms of $\mathbb{A}_i$ that extends ψ and all the descendants of these homomorphisms.*

By calling ELEMENTARY-ENUM$(n, 0, g_0)$ (where g_0 is a trivial mapping from $\emptyset$ to $\mathbb{B}$), we can enumerate all the elementary homomorphisms. By the observation in Lemma 4.2, this means that we can enumerate all the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$.

Theorem 4.7. *For every fixed k, there is a polynomial-delay, polynomial-space algorithm that, given structures $\mathbb{A}$, $\mathbb{B}$, and a sequence of width k endomorphisms of $\mathbb{A}$, enumerates all the homomorphisms from $\mathbb{A}$ to $\mathbb{B}$.*

Theorem 4.7 does not provide a complete description of classes of structures solvable WPD.

Corollary 4.8. *There is a class $\mathcal{A}$ of relational structures such that not all structures from $\mathcal{A}$ have a sequence of width k endomorphisms and* $\mathrm{ECSP}(\mathcal{A}, -)$ *is solvable WPD.*

Proof. Let $\mathcal{A}$ be the class of structures that are the disjoint union of a loop and a core. Obviously, $\mathrm{SCSP}(\mathcal{A}, -)$ is polynomial time solvable. Therefore, by Lemma 3.4, $\mathrm{ECSP}(\mathcal{A}', -)$ is solvable with polynomial delay. However, it is not hard to see that $\mathcal{A}'$ does not have a sequence of endomorphisms of bounded tree width. ∎

Furthermore, as we will see in the next section it is hard, in general, to find a sequence of bounded width endomorphims. Still, we can find a sequence of endomorphisms for a structure $\mathbb{A}$ if we impose two more restrictions on such a sequence.

A retraction φ of a structure $\mathbb{A}$ is called a *k-retraction* if at most k nodes change their value according to φ. A structure is a *k-core* if the only k-retraction is the identity. A k-core of a structure is any k-core obtained by a sequence of k-retractions.

Lemma 4.9. *All k-cores of a structure $\mathbb{A}$ are isomorphic.*

Lemma 4.9 amounts to say that when searching for a sequence of k-retractions converging to a k-core we can use the greedy approach and include, as the next member of such a sequence, any k-retraction with required properties. With this in hands we now can apply Theorem 4.7.

Theorem 4.10. *Let $k > 0$ be a positive integer and let $\mathcal{C}$ be a class of structures such that the k-core of every structure in $\mathcal{C}$ has tree width at most k. Then, the enumeration problem* $\mathrm{ECSP}(\mathcal{C}, -)$ *is solvable WPD.*

Corollary 4.11. *If $\mathcal{C}$ is a class of structures of bounded tree width then* $\mathrm{ECSP}(\mathcal{C}, -)$ *is solvable WPD.*

5. Hardness results

The first result of this section shows that finding a sequence of endomorphisms of bounded width can be difficult even in simplest cases.

Theorem 5.1. *It is NP-complete to decide if a structure has a sequence of 1-width retractions to the core.*

The second result shows that $\mathrm{ECSP}(\mathcal{A}, -)$ can be hard even if every structure in $\mathcal{A}$ has a sequence of width-2 endomorphisms. Note that this result is incomparable with Theorem 5.1, since an enumeration algorithm (in theory) does not necessarily have to compute an sequence of endomorphisms. We need the following lemma:

Lemma 5.2. *If G is a planar graph, then it is possible to find a partition (V_1, V_2) of its vertices in polynomial time such that $G[V_1]$ and $G[V_2]$ have tree width at most 2.*

Proposition 5.3. *There is a class $\mathcal{A}$ of relational structures such that every structure from $\mathcal{A}$ has a sequence of width 2 endomorphisms to the core, and such that the problem $\mathrm{ECSP}(\mathcal{A}, -)$ is not solvable WPD, unless $P = NP$.*

Proof. Let $\mathcal{A}$ be a class of graphs built in the following way. Take a 3-colorable planar graph G and its partition (V_1, V_2) according to Lemma 5.2. Using colorings we can ensure that G is a core. Then we take a disjoint union of this graph with a triangle T having all the colors and a copy G_1 of $G[V_1]$. Let $\mathbb{A}$ denote the resulting structure.

CLAIM 1. $\mathbb{A}$ has a sequence of width-2 endomorphisms.

Let ψ be a 3-coloring of G that is a homomorphism into the triangle, and ψ' the bijective mapping from G_1 to $G[V_1]$. Then φ_1 is defined to act as ψ on G, as ψ' on G'_1 and identically on T. Endomorphism φ_2 is just the 3-coloring of $G \cup G_1$ induced by ψ. The images of φ_1 and φ_2 are $T \cup G[V_1]$ and T, respectively, so all the conditions on a sequence of width-2 homomorphisms are easily checkable.

CLAIM 2. The PLANAR GRAPH 3-COLORING PROBLEM is Turing reducible to $\mathrm{ECSP}(\mathcal{A}, -)$.

Given a planar graph G we find its partition (V_1, V_2) and create a structure $\mathbb{A}$, as described above. Then we apply an algorithm that enumerates solutions to $\mathrm{ECSP}(\mathcal{A}, -)$ We may assume that such an algorithm stops with some time bound regardless whether G is 3-colorable or not. If the algorithm succeeds we can now produce a 3-coloring of G. ∎

6. Conjunctive queries

When making a query to a database one usually needs to obtain values of only those variables (attributes) (s)he is interested in. In terms of homomorphisms this can be translated as follows: For relational structures $\mathbb{A}$, $\mathbb{B}$, and a subset $Y \subseteq A$, we aim to list those mappings from Y to B which can be extended to a full homomorphism from $\mathbb{A}$ to $\mathbb{B}$. In other words, we would like to enumerate all the mappings from Y to B that arise as the restriction of some homomorphism from $\mathbb{A}$ to $\mathbb{B}$. Clearly, this problem significantly differs from the regular enumeration problem. A mapping from Y to B can be extendible to a homomorphism in many ways, possibly superpolynomially many, and an enumeration algorithm would list all of them. In the worst case scenario it would list them before turning to the next partial mapping. If this happens it may destroy polynomiality of the delay between outputting consecutive solutions.

In this section we treat the CONJUNCTIVE QUERY EVALUATION PROBLEM as follows.

CQE($\mathcal{A}, \mathcal{B}$)
Instance: $\mathbb{A} \in \mathcal{A}$, $\mathbb{B} \in \mathcal{B}$, $Y \subseteq A$
Problem: Output all partial mappings from Y to B extendible to a homomorphism from $\mathbb{A}$ to $\mathbb{B}$.

We present two results, first one of them shows that the problem $\mathrm{CQE}(\mathcal{A}, -)$ is WPD when $\mathcal{A}$ is a class of structures of bounded tree width, the second one claims that, modulo some complexity assumptions, in contrast to enumeration problems this cannot be generalized to structures with k-cores of bounded tree width for $k \geq 2$.

Theorem 6.1. *If $\mathcal{A}$ is a class of structures of bounded width then $\mathrm{CQE}(\mathcal{A}, -)$ is solvable WPD.*

Proof. We use Lemma 4.1 to show that algorithm CQE-BOUNDED-WIDTH of Figure 1 does the job. Indeed, this algorithms backtracks only if outputs a solution. ∎

Theorem 6.1 does not generalize to classes of structures whose k-cores have bounded width.

Example 6.2. Recall that the MULTICOLORED CLIQUE problem (cf. [15]) is formulated as follows: Given a number k and a vertex k-colored graph, decide if the graph contains a k-clique all vertices of which are colored different colors. This problem is $W[1]$-complete, i.e., has no time $f(k)n^c$ algorithm for any function f and constant c, unless FPT= $W[1]$. We reduce this problem to $\mathrm{CQE}(\mathcal{A}, -)$ where $\mathcal{A}$ is the class of structures whose 2-cores are 2-element described below.

Let us consider relational structures with two binary and two unary relations. This structure can be thought of as a graph whose vertices and edges have one of the two colors, say, red and blue, accordingly to which of the two binary/unary relations they belong to. Let $\mathbb{A}_k$ be the relational structure with universe $\{a_1, \dots, a_k, y_1, \dots, y_k\}$, where $a_1, \dots, a_k$ are red while $y_1, \dots, y_k$ are blue. Then $\{a_1, \dots, a_k\}$ induces a red clique, that is every a_i, a_j (i, j are not necessarily different) are connected with a red edge, and each y_i is connected to a_i with a blue edge. It is not hard to see that every pair of a red and blue vertices induces a 2-core of this structure. Set $\mathcal{A} = \{\mathbb{A}_k \mid k \in \mathbb{N}\}$.

The reduction of the MULTICOLORED CLIQUE problem to $\mathrm{CQE}(\mathcal{A}, -)$ goes as follows. Given a k-colored graph $G = (V, E)$ whose coloring induces a partition of V into classes $B_1, \dots, B_k$. Then we define structures $\mathbb{A}, \mathbb{B}$ and a set $Y \subseteq A$. We set $\mathbb{A} = \mathbb{A}_k$, $Y = \{y_1, \dots, y_k\}$. Then let $B = V \cup \{b_1, \dots, b_k\}$, the elements of V are colored red and the induced substructure $\mathbb{B}[V]$ is the

Figure 1: Algorithm CQE-BOUNDED-WIDTH

Input: Relational structures $\mathbb{A}$, $\mathbb{B}$, and $Y = \{Y_1, \dots, Y_\ell\} \subseteq A$

Output: A list of mappings $\varphi\colon Y \to B$ extendible to a homomorphism from $\mathbb{A}$ to $\mathbb{B}$

Step 1 **set** $m = 0$, $\varphi = \emptyset$, $S_i = B$, $i \in [m]$, complete:=**false**
Step 2 **while** not complete **do**
Step 2.1 **if** $m < \ell$ **then do**
Step 2.1.1 **search** S_{m+1} until a $b \in S_{m+1}$ is found such that there exists a homomorphism extending $\varphi \cup \{y_{m+1} \to b\}$ and **remove** all members of S_{m+1} preceding b inclusive
Step 2.1.2 **if** such a b exists **then set** $\varphi := \varphi \cup \{y_{m+1} \to b\}$, $m := m + 1$
Step 2.1.3 **else**
Step 2.1.3.1 **if** $m \neq 0$ **then set** $\varphi = \varphi_{|\{y_1,\dots,y_{m-1}\}}$ and $S_{m+1} := B$, $m := m - 1$
Step 2.1.3.2 **else set** complete:=**true**
Step 2.2 **else then do**
Step 2.2.1 **output** φ
Step 2.2.2 **set** $\varphi := \varphi_{|\{y_1,\dots,y_{m-1}\}\}}$, $m := \ell - 1$
endwhile

graph G (without coloring) whose edges are colored also red. Finally, $b_1, \ldots, b_k$ are made blue and each b_i is connected with a blue edge with every vertex from B_i.

It is not hard to see that any homomorphism maps $\{a_1, \ldots, a_k\}$ to V and Y to $\{b_1, \ldots, b_k\}$, and that the number of homomorphisms that do not agree on Y does not exceed k^k. Moreover, G contains a k-colored clique if and only if there is a homomorphism from $\mathbb{A}$ to $\mathbb{B}$ that maps Y onto $\{b_1, \ldots, b_k\}$. If there existed an algorithm solving $\mathrm{CQE}(\mathcal{A}, -)$ WPD, say, time needed to compute the first and every consequent solution is bounded by a polynomial $p(n)$, then time needed to list all solutions is at most $k^k p(n)$. This means that MULTICOLORED CLIQUE is FPT, a contradiction.

References

[1] A. Atserias, A. Bulatov, and V. Dalmau. On the power of *k*-consistency. In *ICALP*, pages 279–290, 2007.

[2] A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. In *FOCS*, 739–748, 2008.

[3] P. Austrin. Towards sharp inapproximability for any 2-csp. In *FOCS*, pages 307–317, 2007.

[4] L. Barto, M. Kozik, and T. Niven. Graphs, polymorphisms and the complexity of homomorphism problems. In *STOC*, pages 789–796, 2008.

[5] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. In *STOC*, pages 29–38, 2008.

[6] A. Bulatov. The complexity of the counting constraint satisfaction problem. In *ICALP*, pages 646–661, 2008.

[7] A. Bulatov and V. Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. In *FOCS*, pages 562–571, 2003.

[8] A. Bulatov and M. Grohe. The complexity of partition functions. *Theoret. Comput. Sci.*, 348:148–186, 2005.

[9] A.A. Bulatov. Tractable conservative constraint satisfaction problems. In *LICS*, pages 321–330, 2003.

[10] A.A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.

[11] A.A. Bulatov, P. Jeavons, and A. A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

[12] H. Chen and V. Dalmau. Beyond Hypertree Width: Decomposition Methods Without Decompositions. In *CP*, pages 167–181, 2005.

[13] M.E. Dyer, L.A. Goldberg, and M. Paterson. On counting homomorphisms to directed acyclic graphs. *J. ACM*, 54(6), 2007.

[14] 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.

[15] M.R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems, 2008. Manuscript.

[16] G. Gottlob, L. Leone, and F. Scarcello. Hypertree decomposition and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002.

[17] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1), 2007.

[18] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *SODA*, pages 289–298, 2006.

[19] P. Hell and J. Nešetril. The core of graph. *Discrete Mathematics*, 109(1-3):117–126, 1992.

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

[21] D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

[22] P. Jonsson, M. Klasson, and A.A. Krokhin. The approximability of three-valued MAX CSP. *SIAM J. Comput.*, 35(6):1329–1349, 2006.

[23] P. Jonsson and A.A. Krokhin. Maximum h-colourable subdigraphs and constraint optimization with arbitrary weights. *J. Comput. Syst. Sci.*, 73(5):691–702, 2007.

[24] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008.

[25] H. Schnoor and I. Schnoor. Enumerating all solutions for constraint satisfaction problems. In *STACS*, pages 694–705, 2007.