Constraint Satisfaction with Succinctly Specified Relations

Hubie Chen and Martin Grohe

[1] Departament de Tecnologia
Universitat Pompeu Fabra
Barcelona, Spain
hubie.chen@upf.edu
[2] Institut für Informatik
Humboldt-Universität Berlin
Unter den Linden 6, D-10099 Berlin, Germany
grohe@informatik.hu-berlin.de

Abstract. The general intractability of the constraint satisfaction problem (CSP) has motivated the study of the complexity of restricted cases of this problem. Thus far, the literature has primarily considered the formulation of the CSP where constraint relations are given explicitly. We initiate the systematic study of CSP complexity with succinctly specified constraint relations.

1. Introduction

Constraint satisfaction problems give a uniform framework for a large number of algorithmic problems in many different areas of computer science, for example, artificial intelligence, database systems, or programming languages. While intractable in general, many restricted constraint satisfaction problems are known to be efficiently solvable. Considerable effort went into analysing the precise conditions that lead to tractable problems; recent results include [2, 8, 3, 16, 7, 6, 1, 5, 17].

An *instance* of a constraint satisfaction problem (CSP) is a triple (V, D, C) consisting of a set V of *variables*, a *domain* D, and a set C of *constraints*. The objective is to find an assignment to the variables, of values from D, such that all constraints in C are satisfied. The constraints are expressions of the form $Rx_1 \ldots x_k$, where R is a k-ary relation on D and $x_1, \ldots, x_k$ are variables. A constraint is satisfied if the k-tuple of values assigned to the variables $x_1, \ldots, x_k$ belongs to the relation R. As a running example for this introduction, let us view SAT, the satisfiability problem for CNF-formulas, as a constraint satisfaction over the domain $\{0, 1\}$. Constraints are given by the clauses of the input formula. For example, the clause $(x \vee \neg y \vee \neg z)$ corresponds to a constraint $Rxyz$, where R is the ternary relation $\{(0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,0,1), (1,1,0), (1,1,1)\}$ on the domain $\{0, 1\}$.

Two types of restriction on CSP-instances are commonly studied in the literature: restrictions of the *constraint language* and *structural restrictions*. The

Dagstuhl Seminar Proceedings 06401
Complexity of Constraints
http://drops.dagstuhl.de/opus/volltexte/2006/802

former restrict the relations on the domain that are permitted in the constraints. For example, HORN-SAT is the restriction of SAT where all constraint relations are specified by Horn clauses, that is, clauses with at most one negative literal. It is known that HORN-SAT can be solved in polynomial time. SAT itself has a restricted constraint language where all constraint relations are specified by disjunctions of literals. The main open problem is the dichotomy conjecture by Feder and Vardi [11], which states that for each constraint language the restricted CSP is either in polynomial time or NP-complete. Currently, this problem still seems to be wide open.

Structural restrictions on CSP-instances are restrictions on the structure induced by the constraints on the variables. A well-known example is the restriction to instances of bounded tree-width. Here a graph on the variables is defined by letting two variables be adjacent if they occur together in some constraint. It is known that for every k the restriction of the general CSP to instances where this graph has tree width at most k, is in polynomial time [10, 13]. The complexity of structural restrictions is better understood than that of constraint language restrictions. If the maximum arity of the constraint relations is bounded, a complete complexity theoretic classification is known [16]; we will state it later in this paper. If the arity is unbounded, interesting classes of tractable problems are known [15, 14, 6, 1, 5, 17], but no complete complexity classification is.

In this paper, we study both restrictions of the constraint language and structural restrictions. We focus on the case of constraint relations of unbounded arity. What is new here is that we pay attention to the way the constraint relations are specified in the problem instances.

In the complexity-theoretic investigations of constraint satisfaction problems it is usually assumed that the constraint relations occurring in a CSP-instance are specified simply by listing all the tuples in the relation, as we did above for the relation specified by the clause $(x \vee \neg y \vee \neg z)$. We call this the *explicit representation.* In practice, the constraint relations are often represented *implicitly.* For example, in SAT-instances, the clauses and not the relations they represent are given. Obviously, the implicit clausal representation is exponentially more succinct than the explicit representation, and this may affect the complexity. As long as the arity of the constraint relations is bounded a priori, as in 3-SAT, it does not make much of difference, because the size of the explicit and any implicit representations differ only in a polynomial factor in terms of the overall instance size. If the domain is fixed, they even differ only by a constant factor. However, for CSPs of unbounded arity, it can make a big difference. What this means in the complexity theoretic context is that algorithms whose running time is polynomial in the size of the explicitly represented instances may become exponential if the instances are represented implicitly. In particular, this is the case for all recent algorithms that exploit a structural restriction called bounded hypertree width and related restrictions [6, 1, 5, 17]. Indeed, these algorithms have been criticised for precisely the reason that they are only polynomial relative to the explicit representation, which is perceived as unrealistic by some researchers. While we do not share this criticism in general, we agree that there are many examples of CSPs where implicit representations are more natural, such as SAT and systems of equalities or inequalities over some numerical domain. This paper initiates a systematic study of the complexity of CSPs with succinctly specified constraint relations.

Before we can state our main results, we have to get a bit more technical.

1.1. Succinctly specified constraint relations. How can we specify constraints implicitly, and how does this affect the complexity of the CSPs? It will be convenient to consider the Boolean domain $\{0, 1\}$ first. An abstract implicit representation is to not specify the constraint relations at all, but just assume a membership oracle for each relation. That is, an algorithm may ask if a specific tuple of values belongs to the relation and get an answer in the next step. However, this may lead to CSPs being highly intractable just because their constraint relations are. Consider the family of CSP-instances $I_n := (\{v_1, \dots, v_n\}, \{0, 1\}, \{R_n v_1 \dots v_n\})$. To solve such instances, the best an algorithm that knows nothing about R_n and only has access to a membership oracle can do is enumerate all tuples in $\{0, 1\}^n$ and query the oracle for each of them. Thus the running time will be exponential in the worst case, even though the instances I_n, having just one constraint, are very simple. This type of complexity is clearly not what we are interested in here. Therefore, specifying the constraint relations by membership oracles is "too implicit"; our implicit representation has to be a bit more explicit. A natural and somewhat generic representation of constraint relations over the Boolean domain is by Boolean circuits. Now consider the family of instances $I_C := (\{v_1, \dots, v_n\}, \{0, 1\}, \{R_C v_1 \dots v_n\})$, where R_C is the n-ary relation specified by the Boolean circuit C with n inputs. Again, this is a family of very simple instances with just one constraint. However, solving the instances in this family amounts to solving the Boolean satisfiability problem, which is NP-complete. *Therefore, it seems reasonable that an implicit representation has a tractable nonemptiness problem.* (The nonemptiness problem for relations specified by circuits is the circuit satisfiability problem.) This not only rules out the representation by arbitrary Boolean circuits, but actually the representation by every class of circuits that contains all CNF-formulas. Thus, in some sense, the generic representation not ruled out by these considerations is the representation by DNF-formulas. This is the representation we shall study on this paper. Of course there are other natural representations that deserve further study, for example, the representation of the constraint relations by ordered binary decision diagrams, but we defer this to future work.

The DNF representation of constraint relations on the Boolean domain has a natural generalisation to arbitrary domains D: We say that a *generalised DNF (GDNF)* representation of a relation $R \subseteq D^k$ is an expression of the form

$$\bigcup_{i=1}^{m} (P_{i1} \times \cdots \times P_{ik}) \qquad (\star)$$

where $P_{ij} \subseteq D$ for $1 \leq i \leq m, 1 \leq j \leq k$. Note that the GDNF enables us to represent relations of size $\Omega(D^k)$ by expressions of size $O(m \cdot |D| \cdot k)$. As the GDNF-representation is the only succinct representation that we study in this paper, from now on we refer to CSPs with constraint relations represented in GDNF as *succinctly represented CSPs (sCSPs).*

Related previous work has studied restrictions on the SAT problem that lead to tractability [19], which in this discussion corresponds to the case where the domain is boolean and the constraint relations are simply given as a disjunction of literals. In contrast, this paper studies a more general representation and does not impose any size restrictions, other than finiteness, on the domain.

1.2. Main Results. We study the complexity of both structural and constraint language restrictions of succinctly represented CSPs.

We give a complete complexity theoretic classification for structural restriction, which generalises the classification for the bounded arity case obtained in [16]. A structural restriction can be described by a class $\mathcal{A}$ of relational structures; we denote the corresponding restricted succinctly represented CSP by $\mathsf{sCSP}(\mathcal{A}, -)$. We prove that, under the complexity theoretic assumption FPT $\neq$ W[1], that $\mathsf{sCSP}(\mathcal{A}, -)$ is in polynomial time if and only if the structures in $\mathcal{A}$ are homomorphically equivalent to structures whose *incidence graph* has bounded tree width (Theorem 8). We refer to this condition by saying that the class $\mathcal{A}$ has *bounded incidence width modulo homomorphic equivalence.*

Constraint language restrictions can also be described by a class $\mathcal{B}$ of structures, and we denote them by $\mathsf{sCSP}(-, \mathcal{B})$. We prove that two general tractability results can be generalised from the explicitly represented to succinctly represented CSPs. These results are formulated in the algebraic language of polymorphisms of the constraint language. We prove that $\mathsf{sCSP}(-, \mathcal{B})$ is in polynomial time if $\mathcal{B}$ is a class of relational structures having a near unanimity polymorphism (Theorem 12), or if $\mathcal{B}$ is a class of relational structures invariant under a set function (Theorem 13); the corresponding results for explicitly represented constraint relations are from [18, 9].

2. Preliminaries, Definitions, and Basic Facts

We use $[n]$ to denote the set containing the first n positive integers, $\{1, \dots, n\}$.

2.1. Relational structures and homomorphisms. As observed by Feder and Vardi [11], constraint satisfaction problems may be viewed as homomorphism problems for relational structures. For the rest of this paper, it will be convenient for us to take this point of view. We review the relevant definitions. A *relational signature* is a finite set of relation symbols, each of which has an associated arity. A *relational structure* $\mathbf{A}$ (over signature σ, for short: σ*-structure*) consists of a universe A and a relation $R^{\mathbf{A}}$ over A for each relation symbol R (of σ), such that the arity of $R^{\mathbf{A}}$ matches the arity associated to R. When $\mathbf{A}$ is a σ-structure and $R \in \sigma$, the elements of $R^{\mathbf{A}}$ are called $\mathbf{A}$-tuples. Throughout this paper, we assume that all relational structures under discussion are finite, that is, have a finite universe. We use boldface letters $\mathbf{A}, \mathbf{B}, \dots$ to denote relational structures, and the corresponding non-boldface letters $A, B, \dots$ to denote their universes. The *arity* of a vocabulary σ is the maximum of the arities of the relation symbols in σ, and the *arity* of a relational structure is the arity of its vocabulary. A class $\mathcal{A}$ of relational structures has *bounded arity* if there is a k such that every structure in $\mathcal{A}$ has arity at most k.

A *substructure* of a relational structure $\mathbf{A}$ is a relational structure $\mathbf{B}$ over the same signature σ as $\mathbf{A}$ where $B \subseteq A$ and $R^{\mathbf{B}} \subseteq R^{\mathbf{A}}$ for all $R \in \sigma$. A *homomorphism* from a relational structure $\mathbf{A}$ to another relational structure $\mathbf{B}$ is a mapping h from the universe of $\mathbf{A}$ to the universe of $\mathbf{B}$ such that for every relation symbol R and every tuple $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$, it holds that $(h(a_1), \ldots, h(a_k)) \in R^{\mathbf{B}}$. (Here, k denotes the arity of R.)

2.2. Explicitly and Succinctly Represented Constraint Satisfaction Problems. With each CSP-instance $I = (V, D, C)$ we associate two relational structures $\mathbf{A}_I$ and $\mathbf{B}_I$ as follows: The signature σ_I of $\mathbf{A}_I$ and $\mathbf{B}_I$ consists of a k-ary relation symbol R for each k-ary constraint relation $R^I \subseteq D^k$ of I. The universe of $\mathbf{B}_I$ is D, and for each relation symbol $R \in \sigma_I$ we let $R^{\mathbf{B}} = R^I$. The universe of $\mathbf{A}_I$ is V, for each k-ary relation symbol $R \in \sigma_I$ we let $R^{\mathbf{A}} = \{(x_1, \ldots, x_k) \mid Rx_1 \ldots x_k \in C\}$. Then a mapping f from $V = A_I$ to $D = B_I$ is a satisfying assignment for I if and only if it is a homomorphism from $\mathbf{A}_I$ to $\mathbf{B}_I$. Thus instance I is satisfiable if and only if there is a homomorphism from $\mathbf{A}_I$ to $\mathbf{B}_I$. Conversely, with every pair $(\mathbf{A}, \mathbf{B})$ of σ-structures we can associate a CSP-instance I such that $\mathbf{A} = \mathbf{A}_I$ and $\mathbf{B} = \mathbf{B}_I$. From now on, we will view CSP-instances as pairs $(\mathbf{A}, \mathbf{B})$ of relational structures of the same signature. For succinctly represented instances, the relations of the structure $\mathbf{B}$ are represented in GDNF.

For all classes $\mathcal{A}, \mathcal{B}$ of structures we let $\mathsf{CSP}(\mathcal{A}, \mathcal{B})$ be the restricted CSP with instances $(\mathbf{A}, \mathbf{B}) \in \mathcal{A} \times \mathcal{B}$. We write $\mathsf{CSP}(-, \mathcal{B})$ or $\mathsf{CSP}(\mathcal{A}, -)$ if $\mathcal{A}$ or $\mathcal{B}$, respectively, is the class of all structures. Constraint language restrictions are restrictions of the form $\mathsf{CSP}(-, \mathcal{B})$, and structural restrictions are restrictions of the form $\mathsf{CSP}(\mathcal{A}, -)$. We write $\mathsf{sCSP}(\mathcal{A}, \mathcal{B})$, $\mathsf{sCSP}(-, \mathcal{B})$, and $\mathsf{sCSP}(\mathcal{A}, -)$ to denote the respective succinctly represented problems.

The following proposition states two simple facts about the relation between explicitly and succinctly represented CSPs.

Proposition 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two classes of relational structures.*

1. $\mathsf{CSP}(\mathcal{A}, \mathcal{B})$ *is polynomial time reducible to* $\mathsf{sCSP}(\mathcal{A}, \mathcal{B})$.
2. *If $\mathcal{A}$ has bounded arity, then* $\mathsf{CSP}(\mathcal{A}, \mathcal{B})$ *and* $\mathsf{sCSP}(\mathcal{A}, \mathcal{B})$ *are polynomial time equivalent.*

Proof. To prove (1), note that a GDNF-representation of a structure $\mathbf{B}$ is obtained by representing each k-ary relation $R^{\mathbf{B}}$ by the expression

$$\bigcup_{(b_1, \ldots, b_k) \in B} (\{b_1\} \times \ldots \times \{b_k\}).$$

Clearly, this GDNF-representation can be computed from the explicit representation in polynomial time.

To prove (2), just note that the explicit representation of a k-ary relation $R^{\mathbf{B}}$ over a domain B can be computed from any GDNF-representation in time

$O(m + |B|^k)$, where m is the size of the GDNF-expression representing $R^{\mathbf{B}}$. If k is bounded by a constant, this is polynomial in the size of the GDNF-representation. □

2.3. Tree Width. A *tree decomposition* of a σ-structure $\mathbf{A}$ is a pair (T, X), where $T = (I, F)$ is a tree, and $X = (X_i)_{i \in I}$ is a family of subsets of A such that for each $R \in \sigma$, say, of arity k, and each $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$ there is a node $i \in I$ such that $\{a_1, \ldots, a_k\} \subseteq X_i$, and for each $a \in A$ the set $\{i \in I \mid a \in X_i\}$ is connected in T. The sets X_i are called the *bags* of the decomposition. The *width* of the decomposition (T, X) is $\max\{|X_i| \mid i \in I\} - 1$, and the *tree width* of $\mathbf{A}$, denoted by $\mathsf{tw}(\mathbf{A})$, is the minimum of the widths of all tree decompositions of $\mathbf{A}$.

2.4. Cores. A *core* of a relational structure $\mathbf{A}$ is a substructure $\mathbf{A}' \subseteq \mathbf{A}$ such that there is a homomorphism from $\mathbf{A}$ to $\mathbf{A}'$, but there is no homomorphism from $\mathbf{A}$ to a proper substructure of $\mathbf{A}'$. We say that a relational structure $\mathbf{A}$ is a *core* if it is its own core. We will make use of the following known and straightforward-to-verify facts concerning cores of finite relational structures: 1) every relational structure $\mathbf{A}$ has a core, 2) any core of a relational structure $\mathbf{A}$ is homomorphically equivalent to $\mathbf{A}$ itself, 3) all cores of a relational structure $\mathbf{A}$ are isomorphic, and 4) a relational structure $\mathbf{A}$ is a core if and only if every homomorphism from $\mathbf{A}$ to $\mathbf{A}$ is surjective. In light of (3), we will use $\mathsf{core}(\mathbf{A})$ to denote a relational structure from the isomorphism class of the cores of $\mathbf{A}$.

The following simple (and known) lemma will be used later:

Lemma 2. *Let $\mathbf{A}$ be a relational structure and $k \geq 1$. Then $\mathbf{A}$ is homomorphically equivalent to a relational structure of tree width at most k if and only if $\mathsf{tw}(\mathsf{core}(\mathbf{A})) \leq k$.*

2.5. Previous Complexity Results. A class $\mathcal{A}$ of structures has *bounded tree width* if there is a k such that every structure in $\mathcal{A}$ has tree width at most k. The class $\mathcal{A}$ has *bounded tree width modulo homomorphic equivalence* if there is a k such that every structure in $\mathcal{A}$ is homomorphically equivalent to a structure of tree width at most k.

We will make use of the following previously established results on structural tractability.

Theorem 3 (Dalmau, Kolaitis, and Vardi [8]). *Let $\mathcal{A}$ be a class of relational structures. If $\mathcal{A}$ has bounded tree width modulo homomorphic equivalence, then $\mathsf{CSP}(\mathcal{A}, -)$ is in polynomial time.*

Theorem 4 (Grohe [16]). *Assume that* FPT $\neq$ W[1]. *Let $\mathcal{A}$ be a recursively enumerable class of relational structures of bounded arity. If $\mathsf{CSP}(\mathcal{A}, -)$ is in polynomial time, then $\mathcal{A}$ has bounded tree width modulo homomorphic equivalence.*

Note that FPT and W[1] are two complexity classes (from parameterized complexity theory) that are believed to be distinct. It is not necessary for this paper to know about these classes.

The assumption that $\mathcal{A}$ be recursively enumerable in the last theorem is inessential and can be dropped if the complexity theoretic assumption FPT $\neq$ W[1] is replaced by a slightly stronger assumption. Then for classes $\mathcal{A}$ of bounded arity, the combination of the two theorems completely characterises the tractable structural restrictions. There are classes of unbounded arity that are not of bounded tree width modulo homomorphic equivalence, but still have a tractable CSP. Examples are all classes that have *bounded generalised hypertree width modulo homomorphic equivalence* [5].

3. Structural Restrictions

The goal of this section is to generalise the characterisation of tractable structural restrictions of explicitly represented CSPs of bounded arity provided by Theorems 3 and 4 to succinctly represented CSPs (of possibly unbounded arity). First observe that the theorems can immediately transfered to succinctly represented CSPs in the bounded arity case, simply because for all classes $\mathcal{A}$ of bounded arity the problems $\mathsf{sCSP}(\mathcal{A}, -)$ and $\mathsf{CSP}(\mathcal{A}, -)$ are polynomial time equivalent (Proposition 1). Thus we obtain the following corollary of Theorems 3 and 4:

Corollary 5. *Assume that* FPT $\neq$ W[1]. *Let $\mathcal{A}$ be a recursively enumerable class of relational structures of bounded arity. Then* $\mathsf{sCSP}(\mathcal{A}, -)$ *is in polynomial time if and only if $\mathcal{A}$ has bounded tree width modulo homomorphic equivalence.*

3.1. Incidence Width.

Definition 6. *1. The incidence signature* $\mathsf{inc}(\sigma)$ *of a relational signature σ contains k relation symbols $R_1, \ldots, R_k$ of arity two for every relation symbol R of σ having arity k.*

2. Let $\mathbf{A}$ be a relational structure over signature σ. The incidence structure $\mathsf{inc}(\mathbf{A})$ *of $\mathbf{A}$ is the relational structure over signature* $\mathsf{inc}(\sigma)$

- *having universe $A \cup \bigcup_{R\in\sigma}\{(R, a_1, \ldots, a_k) : (a_1, \ldots, a_k) \in R^{\mathbf{A}}\}$, and*
- *where for each relation symbol R of σ having arity k, we define*

$$R_i^{\mathsf{inc}(\mathbf{A})} = \{((R, a_1, \ldots, a_k), a_i) : (a_1, \ldots, a_k) \in R^{\mathbf{A}}\}$$

for all $i \in [k]$.

Note that the incidence structure $\mathsf{inc}(\mathbf{A})$ of a structure $\mathbf{A}$ is a binary structure that carries the same information as $\mathbf{A}$. It also has about the same size, if we count as the size of a structure as the size of the universe plus the size of all tuples in all relations. However, the incidence structure can have much smaller tree width: If $\mathbf{A}$ is a structure with universe $[n]$ and one relation $R_n^{\mathbf{A}}$ that only contains the tuple $(1, \ldots, n)$, then $\mathsf{tw}(\mathbf{A}) = n - 1$ and $\mathsf{tw}(\mathsf{inc}(\mathbf{A})) = 1$.

Definition 7. *The* incidence width $\mathsf{iw}(\mathbf{A})$ *of a relational structure $\mathbf{A}$ is the tree width of its incidence structure, that is,*

$$\mathsf{iw}(\mathbf{A}) = \mathsf{tw}(\mathsf{inc}(\mathbf{A})).$$

The measure of incidence width has been previously studied (e.g., [12, 19]). It is easy to see that for every structure $\mathbf{A}$ we have

$$\mathsf{iw}(\mathbf{A}) \leq \mathsf{tw}(\mathbf{A}) + 1.$$

The example given right before the definition of incidence width shows that the inequality can be strict.

3.2. Characterisation of tractable restrictions. A class $\mathcal{A}$ of structures has *bounded incidence width modulo homomorphic equivalence* if there is a k such that every structure in $\mathcal{A}$ is homomorphically equivalent to a structure of incidence width at most k.

Theorem 8. *Assume that* $\mathrm{FPT} \neq \mathrm{W}[1]$. *Let* $\mathcal{A}$ *be a recursively enumerable class of relational structures. Then* $\mathsf{sCSP}(\mathcal{A}, -)$ *is in polynomial time if and only if* $\mathcal{A}$ *has bounded incidence width modulo homomorphic equivalence.*

The rest of this section is devoted to a proof of this theorem. We require some more preparatory lemmas. The proofs of Lemmas 9 and 10 can be found in the appendix.

Lemma 9. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be relational structures over the same signature, and let* h *be a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$.

- *There is a unique extension* h' *of* h *that is a homomorphism from* $\mathsf{inc}(\mathbf{A})$ *to* $\mathsf{inc}(\mathbf{B})$*, given by*

$$h'(R, a_1, \dots, a_k) = (R, h(a_1), \dots, h(a_k))$$

  for all tuples $(R, a_1, \dots, a_k)$ *in the universe of* $\mathsf{inc}(\mathbf{A})$.
- *The restriction to* A *of any homomorphism from* $\mathsf{inc}(\mathbf{A})$ *to* $\mathsf{inc}(\mathbf{B})$ *is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$.

Lemma 10. *For any relational structure* $\mathbf{A}$*, it holds that the relational structures* $\mathsf{core}(\mathsf{inc}(\mathbf{A}))$ *and* $\mathsf{inc}(\mathsf{core}(\mathbf{A}))$ *are isomorphic.*

For a class $\mathcal{A}$ of structures, we let $\mathsf{inc}(\mathcal{A}) = \{\mathsf{inc}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{A}\}$ and similarly $\mathsf{core}(\mathcal{A}) = \{\mathsf{core}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{A}\}$.

Lemma 11. *For every class* $\mathcal{A}$ *of relational structures, the following four statements are equivalent:*

1. $\mathcal{A}$ *has bounded incidence width modulo homomorphic equivalence.*
2. $\mathsf{core}(\mathcal{A})$ *has bounded incidence width.*
3. $\mathsf{core}(\mathsf{inc}(\mathcal{A}))$ *has bounded tree width.*
4. $\mathsf{inc}(\mathcal{A})$ *has bounded tree width modulo homomorphic equivalence.*

Proof. Follows immediately from Lemmas 2 and 10. □

Proof (of Theorem 8). The idea of the proof is to give reductions between the problems $\mathsf{sCSP}(\mathcal{A}, -)$ and $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$. For the forward direction, suppose that $\mathcal{A}$ has bounded incidence width modulo homomorphic equivalence. Then by Lemma 11, $\mathsf{inc}(\mathcal{A})$ has bounded tree width modulo homomorphic equivalence. Thus by Theorem 3, $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ is in polynomial time. We show that $\mathsf{sCSP}(\mathcal{A}, -)$ is in polynomial time by giving a polynomial-time reduction to $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$.

We reduce an instance $(\mathbf{A}, \mathbf{B})$ of $\mathsf{sCSP}(\mathcal{A}, -)$, where $\mathbf{B}$ is represented succinctly, to an instance $(\mathsf{inc}(\mathbf{A}), \mathbf{B}')$ of $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ for a relational structure $\mathbf{B}'$ to be defined next. Before we define $\mathbf{B}'$, note that we cannot simply let $\mathbf{B}' = \mathsf{inc}(\mathbf{B})$, because $\mathsf{inc}(\mathbf{B})$, having roughly the same size as $\mathbf{B}$ represented explicitly, can be exponentially larger than the succinct GDNF-representation of $\mathbf{B}$ and hence cannot be constructed in polynomial time. Let us turn to the definition of $\mathbf{B}'$. Let $\sigma = \{R_1, \dots, R_\ell\}$, where R_i is r_i-ary, be the signature of $\mathbf{A}$ and $\mathbf{B}$. Suppose that, for $1 \le i \le \ell$, the GDNF representation of $R_i^{\mathbf{B}}$ is

$$\bigcup_{j=1}^{m_i} (P_{ij1} \times \ldots \times P_{ijr_i}),$$

where $P_{ijk} \subseteq B$.

- The signature of $\mathbf{B}'$ is $\mathsf{inc}(\sigma)$.
- The universe of $\mathbf{B}'$ is

$$B' = B \cup \{p_{ij} \mid 1 \le i \le \ell, 1 \le j \le m_i\},$$

  where the p_{ij} are new elements not contained in B.
- For $1 \le i \le m$, the binary relations $R_{i1}^{\mathbf{B}'}, \dots, R_{ir_i}^{\mathbf{B}'}$ are defined by

$$R_{ik}^{\mathbf{B}'} = \{(p_{ij}, b) \mid 1 \le j \le m_i, b \in P_{ijk}\}$$

  for $1 \le k \le r_i$.

Note that $\mathbf{B}'$ can be constructed from the succinct representation of $\mathbf{B}$ in polynomial time. Thus it suffices to prove that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if there is a homomorphism from $\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$.

Let h be a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Let h' be an extension of h where, for every element $(R_i, a_1, \dots, a_{r_i})$ in the universe of $\mathsf{inc}(\mathbf{A})$, $h'(R_i, a_1, \dots, a_{r_i})$ is defined to be an element p_{ij} for some $j \in [m_i]$ with $(h(a_i), \dots, h(a_{r_i})) \in P_{ij1} \times \ldots \times P_{ijr_i}$. Such a j exists, because

$$(h(a_i), \dots, h(a_{r_i})) \in R_i^{\mathbf{B}} = \bigcup_{j=1}^{m_i} (P_{ij1} \times \ldots \times P_{ijr_i}).$$

It is straightforward to verify that h' is a homomorphism from $\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$. Furthermore, it is straightforward to verify that if h' is a homomorphism from

$\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$, then the restriction of h' to A is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. This completes the proof of the forward direction of Theorem 8.

For the backward direction, suppose that $\mathcal{A}$ does not have bounded incidence-width modulo homomorphic equivalence. We wish to show that $\mathsf{sCSP}(\mathcal{A}, -)$ is not in polynomial time. By Lemma 11, $\mathsf{inc}(\mathcal{A})$ does not have bounded tree width modulo homomorphic equivalence. Noting that the recursive enumerability of $\mathcal{A}$ implies the recursive enumerability of $\mathsf{inc}(\mathcal{A})$ and that $\mathsf{inc}(\mathcal{A})$ is binary, $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ is not in polynomial time by Theorem 4. Thus it suffices to give a polynomial-time reduction from $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ to $\mathsf{sCSP}(\mathcal{A}, -)$. Given an instance $(\mathsf{inc}(\mathbf{A}), \mathbf{B}')$ of $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$, we create an equivalent instance $(\mathbf{A}, \mathbf{B})$ of $\mathsf{sCSP}(\mathcal{A}, -)$. Let σ be the signature of $\mathbf{A}$. Then $\mathsf{inc}(\mathbf{A})$ and $\mathbf{B}'$ have signature $\mathsf{inc}(\sigma)$. Without loss of generality we may assume that $\mathbf{A}$ has no isolated vertices, that is, every $a \in A$ is contained in some tuple in some relation of $\mathbf{A}$. We can make this assumption because isolated vertices can be mapped anywhere by a homomorphism and thus are not relevant when it comes to the existence of a homomorphism.

Let B be the set of all $b \in B'$ such that there exists an $R \in \sigma$, say, of arity k, an $i \in [k]$, and a $b' \in B'$ such that $(b', b) \in R_i^{\mathbf{B}}$. For every relation symbol $R \in \sigma$ of arity k, let $T_R = \bigcap_{i \in [k]} \{b' \in B' : (b', b) \in R_i^{\mathbf{B}} \text{ for some } b \in B'\}$. If $\mathbf{B}'$ were of the form $\mathsf{inc}(\mathbf{B}'')$ for some σ-structure $\mathbf{B}''$, then the universe of $\mathbf{B}''$ would be B, and the elements of T_R would represent the tuples in $R^{\mathbf{B}''}$, that is, we would have $T_R = \{(R, b_1, \ldots, b_k) \mid (b_1, \ldots, b_k) \in R^{\mathbf{B}''}\}$. But $\mathbf{B}'$ is not necessarily $\mathsf{inc}(\mathbf{B}'')$ for any $\mathbf{B}''$. However, every homomorphism h from $\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$ must map all elements of A to elements of B and all elements of the form $(R, a_1, \ldots, a_k)$ to elements of T_R. The former holds because $\mathbf{A}$ has no isolated vertices, and the latter because for all $a' = (R, a_1, \ldots, a_k)$ it holds that $a' \in \bigcap_{i \in [k]} \{a'' : (a'', a) \in R_i^{\mathsf{inc}(\mathbf{A})} \text{ for some } a\}$.

For every k-ary $R \in \sigma$, $b \in T_R$, and $i \in [k]$ we let $P_{Rbi} = \{b' \in B \mid (b, b') \in R_i^{\mathbf{B}'}\}$. We define $\mathbf{B}$ to be the structure with universe B and, for k-ary $R \in \sigma$,

$$R^{\mathbf{B}} = \bigcup_{b \in T_R} (P_{Rb1} \times \ldots \times P_{Rbk}). \qquad (\star)$$

It is easy to see that if h is a homomorphism from $\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$, then the restriction of h to A is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ and that, conversely, every homomorphism from $\mathbf{A}$ to $\mathbf{B}$ can be extended to a homomorphism from $\mathsf{inc}(\mathbf{A})$ to $\mathbf{B}'$. Furthermore, the succinct representation of $\mathbf{B}$, where the relations are represented by the GDNF-expressions on the right hand side of $(\star)$, can be computed from $\mathbf{B}'$ in polynomial time. $\square$

4. Constraint language restrictions

This section presents a pair of tractability results based on constraint language restrictions. The first is based on *near-unanimity polymorphisms* which were studied in the CSP (with explicitly represented tuples) in [18]. A *near-unanimity* *operation* is an operation $f : D^k \to D$ of arity $k \geq 3$ satisfying the identities

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

An operation $f : B^k \to B$ is a *polymorphism* of a relational structure $\mathbf{B}$ if it is a homomorphism from $\mathbf{B}^k$ to $\mathbf{B}$. Let us recall that, for a relational structure $\mathbf{B}$, the relational structure $\mathbf{B}^k$ is the structure with universe B^k and where $R^{\mathbf{B}^k}$ is defined as

$$\{((b_{11}, \ldots, b_{1k}), \ldots, (b_{m1}, \ldots, b_{mk})) : (b_{11}, \ldots, b_{m1}), \ldots, (b_{1k}, \ldots, b_{mk}) \in R^{\mathbf{B}}\}$$

for all relation symbols R of arity m.

Theorem 12. *Let $\mathcal{B}_k$ be the set of all succinctly specified relational structures having a near-unanimity polymorphism of arity k. For each $k \geq 3$, the problem* $\mathsf{sCSP}(-, \mathcal{B}_k)$ *is in polynomial time.*

A proof sketch can be found in Appendix A.3.

When $\mathbf{B}$ is a relational structure over signature σ, we define $\mathcal{P}(\mathbf{B})$ to be the relational structure having universe $\wp(B) \setminus \{\emptyset\}$ and where for each $R \in \sigma$ of arity k, the relation $R^{\mathcal{P}(\mathbf{B})}$ is defined as $\{(\mathsf{pr}_1 S, \ldots, \mathsf{pr}_k S) : S \subseteq R^{\mathbf{B}}, S \neq \emptyset\}$. Here, $\wp(B)$ denotes the power set of B, and for a set S of k-tuples, and $i \in [k]$, $\mathsf{pr}_i S$ denotes the set $\{b_i : (b_1, \ldots, b_k) \in S\}$. We say that $\mathbf{B}$ is *invariant under a set function* if there exists a homomorphism from $\mathcal{P}(\mathbf{B})$ to $\mathbf{B}$. In the context of constraint satisfaction problems, set functions have been studied in [9].

Theorem 13. *Let $\mathcal{B}$ be the set of all succinctly specified relational structures invariant under a set function. The problem* $\mathsf{sCSP}(-, \mathcal{B})$ *is in polynomial time.*

A proof can be found in Appendix A.4.

5. Conclusions

We have initiated a study of the complexity of succinctly represented constraint satisfaction problems. We believe that it is worthwhile to look at succinct representations, because important examples of constraint satisfaction problems are usually specified succinctly. Our results are not deep, but in particular the complete characterization of tractable structural restrictions is is quite nice (and surprisingly simple and clean cut). Note that no corresponding classification result is known for explicitly represented CSPs.

In this paper, we have only looked at one specific succinct representation, the generalized DNF representations. There are other natural candidates; in particular, the representation of the constraint relations by ordered binary decision diagrams and their natural generalizations from the Boolean to other domains seem worth being studied.

References

1. I. Adler, G. Gottlob, and M. Grohe. Hypertree-width and related hypergraph invariants. In S. Felsner, editor, *Proceedings of the 3rd European Conference on Combinatorics, Graph Theory, and Applications*, volume AE of *DMTCS Proceedings Series*, pages 5–10, 2005.
2. Andrei Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of 43rd IEEE Symposium on Foundations of Computer Science*, pages 649–658, 2002.
3. Andrei Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of 18th IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 321–330, 2003. Extended version appears as Oxford University technical report PRG-RR–03-01.
4. Hubie Chen and Victor Dalmau. (Smart) look-ahead arc consistency and the pursuit of CSP tractability. In *Principles and Practice of Constraint Programming - CP 2004*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
5. Hubie Chen and Victor Dalmau. Beyond hypertree width: Decomposition methods without decompositions. In *CP 2005*, 2005.
6. D. Cohen, P. Jeavons, and M. Gyssens. A unified theory of structural tractability for constraint satisfaction and spread cut decomposition. IJCAI 2005, 2005.
7. Victor Dalmau. Mal'tsev constraints made simple. ECCC technical report, 2004.
8. Victor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *Constraint Programming '02*, LNCS, 2002.
9. Victor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *CP 1999*, pages 159–173, 1999.
10. Rina Dechter and Judea Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, pages 353–366, 1989.
11. Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.
12. J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. *Journal of the ACM*, 49(6):716–752, 2002.
13. Eugene Freuder. Complexity of k-tree structured constraint satisfaction problems. In *AAAI-90*, 1990.
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. Georg Gottlob, Nicola Leone, and Francesco Scarcello. A comparison of structural csp decomposition methods. *Artif. Intell.*, 124(2):243–282, 2000.
16. Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. In *FOCS 2003*, pages 552–561, 2003.
17. Martin Grohe and Daniel Marx. Constraint solving via fractional edge covers. In *Proceedings of the of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, 2006.
18. Peter Jeavons, David Cohen, and Martin Cooper. Constraints, consistency, and closure. *Articial Intelligence*, 101(1-2):251–265, 1998.
19. Stefan Szeider. On fixed-parameter tractable parameterizations of sat. *Theory and Applications of Satisfiability (Selected and Revised Papers of SAT 2003), Lecture Notes in Computer Science 2919*, pages 188–202, 2004.

A. Appendix

A.1. Proof of Lemma 9.

Proof. We begin with the first claim. It is straightforward to show that the extension h' is a homomorphism, so we prove its uniqueness. Let g be any homomorphism from $\mathsf{inc}(\mathbf{A})$ to $\mathsf{inc}(\mathbf{B})$ extending h. Let $(R, a_1, \ldots, a_k)$ be a tuple from the universe of $\mathsf{inc}(\mathbf{A})$. For every i, we have $((R, a_1, \ldots, a_k), a_i) \in R_i^{\mathsf{inc}(\mathbf{A})}$; since the projection of $R_i^{\mathsf{inc}(\mathbf{B})}$ onto the first coordinate only contains tuples of the form $(R, b'_1, \ldots, b'_k)$ where $(b'_1, \ldots, b'_k) \in R^{\mathbf{B}}$, we have $g(R, a_1, \ldots, a_k) = (R, b_1, \ldots, b_k)$ where $(b_1, \ldots, b_k) \in R^{\mathbf{B}}$. Since g is a homomorphism extending h, we have $(g(R, a_1, \ldots, a_k), g(a_i)) = ((R, b_1, \ldots, b_k), h(a_i))$. By the definition of $\mathsf{inc}(\mathbf{B})$, we have $h(a_i) = b_i$. This argument holds for all i, so we conclude that $g(R, a_1, \ldots, a_k) = (R, h(a_1), \ldots, h(a_k))$.

Now we prove the second claim. Let $h' : \mathsf{inc}(\mathbf{A}) \to \mathsf{inc}(\mathbf{B})$ be a homomorphism. Let $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$. We want to show that $(h'(a_1), \ldots, h'(a_k)) \in R^{\mathbf{B}}$. As in the proof of the first claim, we have $h'(R, a_1, \ldots, a_k) = (R, b_1, \ldots, b_k)$ for some tuple $(b_1, \ldots, b_k) \in R^{\mathbf{B}}$. For all $i \in [k]$, we have $((R, a_1, \ldots, a_k), a_i) \in R_i^{\mathsf{inc}(\mathbf{A})}$; mapping this tuple under h', we obtain $((R, b_1, \ldots, b_k), h'(a_i)) \in R_i^{\mathsf{inc}(\mathbf{B})}$. By definition of $\mathsf{inc}(\mathbf{B})$, we have that $h'(a_i) = b_i$, so $(h'(a_1), \ldots, h'(a_k)) = (b_1, \ldots, b_k) \in R^{\mathbf{B}}$. □

A.2. Proof of Lemma 10.

Proof. The structures $\mathbf{A}$ and $\mathsf{core}(\mathbf{A})$ are homomorphically equivalent; by use of Lemma 9, it follows that the structures $\mathsf{inc}(\mathbf{A})$ and $\mathsf{inc}(\mathsf{core}(\mathbf{A}))$ are homomorphically equivalent. To establish the desired isomorphism, it suffices to show that the structure $\mathsf{inc}(\mathsf{core}(\mathbf{A}))$ is a core.

Let $\mathbf{C}$ be a core of $\mathbf{A}$. Let $h' : \mathsf{inc}(\mathbf{C}) \to \mathsf{inc}(\mathbf{C})$ be a homomorphism. We claim that h' is surjective, which suffices. Let h be the restriction of h' to the universe C of $\mathbf{C}$. We have that h is surjective onto C since $\mathbf{C}$ is a core. Also, since $\mathbf{C}$ is a core, for every $\mathbf{C}$-tuple $(c_1, \ldots, c_k) \in R^{\mathbf{C}}$ we have that there exists another $\mathbf{C}$-tuple $(b_1, \ldots, b_k) \in R^{\mathbf{C}}$ such that $(h(b_1), \ldots, h(b_k)) = (c_1, \ldots, c_k)$. It follows by Lemma 9 that for every element in the universe of $\mathsf{inc}(\mathbf{C})$ of the form $(R, c_1, \ldots, c_k)$, there exists an element $(R, b_1, \ldots, b_k)$ also in the universe of $\mathsf{inc}(\mathbf{C})$ such that $h'(R, b_1, \ldots, b_k) = (R, c_1, \ldots, c_k)$, implying that h' is surjective as desired. □

A.3. Proof of Theorem 12.

Proof (idea). We give a reduction from $\mathsf{sCSP}(-, \mathcal{B})$ to $\mathsf{CSP}(-, \mathcal{B})$; the latter is tractable by the "strong k-consistency" algorithm, see [18] for a description of this algorithm and proof. Note that this algorithm, for any fixed k, runs in polynomial time. For ease of notation, we describe the reduction using the definition of CSP given in the introduction.

Let (V, D, C) be an instance of $\mathsf{sCSP}(-, \mathcal{B})$; we create an instance (V', D', C') of $\mathsf{CSP}(-, \mathcal{B})$ as follows. Set $V' = V$ and $D' = D$. For each constraint $Rx_1 \ldots x_n \in$

C, we create constraints in C' as follows. Let $\bigcup_{i=1}^{m}(P_{i1}\times\cdots\times P_{in})$ denote the representation of R. If $n \leq k-1$, then we simply place $R'x_1 \ldots x_n$ in C', where R' is the explicit representation of R, that is, the set of all tuples $\bigcup_{i=1}^{m}(P_{i1}\times\cdots\times P_{in})$. If $n \geq k$, then for each sequence $j_1, \ldots, j_l \in [n]$ where $j_1 < \cdots < j_l$ and $l = k-1$, we create an arity l constraint $R_{j_1,\ldots,j_l}x_{j_1}\ldots x_{j_l}$ where $R_{j_1,\ldots,j_l}$ is the explicit representation of the tuples $\bigcup_{i=1}^{m}(P_{ij_1}\times\cdots\times P_{ij_l})$.

Clearly, any solution to (V, D, C) is also a solution to (V', D', C'). However, since the relations of (V, D, C) have a near-unanimity polymorphism of arity k, the relations are $(k-1)$-decomposable by [18], implying that any solution to (V', D', C') is also a solution to (V, D, C). Moreover, polynomially many constraints are placed in C', and each of these have arity bounded above by $k-1$, so it is straightforward to verify that (V', D', C') can be computed from (V, D, C) in polynomial time. □

A.4. Proof of Theorem 13.

Proof. We show how to implement the *arc consistency* procedure on succinctly represented instance. This is a well known procedure for explicitly represented CSPs that are invariant under a set function; see the papers [9, 4] for more information on arc consistency and tractability.

Let $(\mathbf{A}, \mathbf{B})$ be an instance of $\mathsf{sCSP}(-, \mathcal{B})$, and let σ be the signature of the structures. Without loss of generality we may assume that every relation $R^{\mathbf{A}}$ contains exactly one tuple. To achieve this, suppose that $R^{\mathbf{A}} = \{\bar{a}_1, \ldots, \bar{a}_m\}$ for some $m \geq 2$. Let $R_1, \ldots, R_m$ be new relation symbols of the same arity as R. Let $\sigma' = (\sigma \setminus \{R\}) \cup \{R_1, \ldots, R_m\}$, and let $\mathbf{A}'$ be the σ'-structure obtained from $\mathbf{A}$ by letting $R_i^{\mathbf{A}'} = \{\bar{a}_i\}$ for $1 \leq i \leq m$ and $S^{\mathbf{A}'} = S^{\mathbf{A}}$ for all $S \in \sigma \setminus \{R\}$. Let $\mathbf{B}'$ be the σ'-structure obtained from $\mathbf{B}$ by letting $R_i^{\mathbf{B}'} = R^{\mathbf{B}}$ for $1 \leq i \leq m$ and $S^{\mathbf{B}'} = S^{\mathbf{B}}$ for all $S \in \sigma \setminus \{R\}$. Then obviously a mapping $h : A \to B$ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if it is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$, and $(\mathbf{A}', \mathbf{B}')$ can be computed from $(\mathbf{A}, \mathbf{B})$ in polynomial time. Repeating this for all $S \in \sigma$ with $|S^{\mathbf{A}}| \geq 2$, we get an instance with the desired property. So in the following, we assume that $|R^{\mathbf{A}}| = 1$ for all $R \in \sigma$.

For every $a \in A$, let $O(a)$ be the of all *occurrences* of a, that is, all pairs (R, j) such that a occurs on the jth position of the tuple in $R^{\mathbf{A}}$. The algorithm repeatedly carries out the following procedure: It first computes the intersection

$$I(a, \mathbf{B}) = \bigcap_{(R,j)\in O(a)} \mathsf{pr}_j(R^{\mathbf{B}}),$$

where $\mathsf{pr}_j(R^{\mathbf{B}})$ denotes the projection of $R^{\mathbf{B}}$ on the jth position. Clearly, every homomorphism from $\mathbf{A}$ to $\mathbf{B}$ must map a to an element of $I(a, \mathbf{B})$. If $I(a, \mathbf{B}) = \emptyset$, then there is no homomorphism from $\mathbf{A}$ to $\mathbf{B}$; we say that the algorithm has *detected an inconsistency.* Otherwise, for each occurrence $(R, j) \in O(a)$, the algorithm removes from $R^{\mathbf{B}}$ all tuples $(b_1, \ldots, b_k)$ such that $b_j \notin I(a, \mathbf{B})$. Let $\mathbf{B}'$ be the resulting structure. Then there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}'$, because tuples $(b_1, \ldots, b_k) \in R^{\mathbf{B}}$

with $b_j \notin I(a, \mathbf{B})$ can never appear in the image of a homomorphism anyway. Observe that a succinct representation of $\mathbf{B}'$ can be computed from a succinct representation of $\mathbf{B}$ in polynomial time as follows: If the GDNF-representation of $R^{\mathbf{B}}$ is

$$\bigcup_{i=1}^{m} (P_{i1} \times \cdots \times P_{ik}),$$

then a GDNF-representation of $R^{\mathbf{B}'}$ is

$$\bigcup_{i=1}^{m} (P'_{i1} \times \cdots \times P'_{ik}),$$

where

$$P'_{ij} = \begin{cases} P_{ij} \cap I(a, \mathbf{B}) & \text{if } (R, j) \in O(a), \\ P_{ij} & \text{otherwise.} \end{cases}$$

Now this whole procedure is repeated on $(\mathbf{A}, \mathbf{B}')$ for some $a' \in A$, and then on the resulting instance $(\mathbf{A}, \mathbf{B}'')$, et cetera, until either an inconsistency is detected or *consistency is established*, that is, we have reached a structure $\mathbf{B}^*$ such that for all $a \in A$ and $(R, j) \in O(a)$ we have

$$\mathsf{pr}_j(R^{\mathbf{B}^*}) = I(a, \mathbf{B}^*). \tag{1}$$

(In this situation, any further applications of the procedure would yield the same $\mathbf{B}^*$.) It is clear that the number of iterations needed to establish consistency is polynomial in the size of the instance, because in each iteration an element of a set in the GDNF of the structure $\mathbf{B}$ is removed, and there are polynomially many such elements.

Suppose now that no inconsistency is detected and that consistency is established and let $(\mathbf{A}, \mathbf{B}^*)$ be the resulting instance. Note that $(\mathbf{A}, \mathbf{B}^*)$ is equivalent to the original instance and that (1) holds for all $a \in A$ and $(R, j) \in O(a)$. We claim that the mapping h defined

$$h(a) = I(a, \mathbf{B}^*)$$

is a homomorphism from $\mathbf{A}$ to $\mathcal{P}(\mathbf{B})$. Since $\mathbf{B}^*$ is a substructure of $\mathbf{B}$ and therefore $I(a, \mathbf{B}^*) \subseteq B^* \subseteq B$, and $I(a, \mathbf{B}^*) \neq \emptyset$ for all $a \in A$, h is indeed a mapping from A to $\wp(B) \setminus \emptyset$. To prove that it is a homomorphism, let $(a_1, \dots, a_k) \in R^{\mathbf{A}}$ for some k-ary $R \in \sigma$. Then $(R, j) \in O(a_j)$ for all $j \in [k]$. Thus, by (1), we have $\mathsf{pr}_j(R^{\mathbf{B}^*}) = I(a_j, \mathbf{B}^*)$. As $\mathbf{B}^* \subseteq \mathbf{B}$, we have $R^{\mathbf{B}^*} \subseteq R^{\mathbf{B}}$ and thus, by the definition of $\mathcal{P}(\mathbf{B})$,

$$\big(h(a_1), \dots, h(a_k)\big) = \big(\mathsf{pr}_1(R^{\mathbf{B}^*}), \dots, \mathsf{pr}_k(R^{\mathbf{B}^*})\big) \in R^{\mathcal{P}(\mathbf{B})}.$$

This proves that h is a homomorphism. Composing the homomorphism h with a homomorphism from $\mathcal{P}(\mathbf{B})$ to $\mathbf{B}$, we have a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. □