Constraint Satisfaction with Succinctly Specified Relations

Hubie Chen[a], Martin Grohe[b]

[a] *Dept. de Tecnologies de la Informació i les Comunicacions*
Universitat Pompeu Fabra
Passeig de Circumval·lació, 8
08003 Barcelona, Spain
[b] *Institut für Informatik*
Humboldt-Universität Berlin
Unter den Linden 6
D-10099 Berlin, Germany

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 [1, 2, 3, 4, 5, 6, 7, 8, 9].

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 $Rx_1 \dots x_k$, where R is a k-ary relation on D and $x_1, \dots, x_k$ are variables. A constraint is satisfied if the k-tuple of values assigned to the variables $x_1, \dots, x_k$ 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 problem 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 $\{0, 1\}$.

Email addresses: hubie.chen@upf.edu (Hubie Chen), grohe@informatik.hu-berlin.de (Martin Grohe)

Preprint submitted to Elsevier *April 15, 2010*

Two types of restriction on CSP-instances are commonly studied in the literature: restrictions of the *constraint language* and *structural restrictions*. The 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 [10], 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 [11, 12]. 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 [4]; we will state it later in this paper. If the arity is unbounded, interesting classes of tractable problems are known [13, 14, 6, 7, 8, 9], but no complete complexity classification is.

In this paper, we study both structural restrictions and restrictions of the constraint language. 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 representation 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, 7, 8, 9]. 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. We investigate two different succinct representations.

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 $I_n := (\{v_1, \ldots, v_n\}, \{0, 1\}, \{R_n v_1 \ldots v_n\})$. 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

$$I_C := (\{v_1, \ldots, v_n\}, \{0, 1\}, \{R_C v_1 \ldots v_n\}),$$

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. *From these examples, it seems reasonable to assume 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 terms of circuits, the generic representation not ruled out by these considerations is the representation by DNF-formulas. This is the first succinct representation we shall study on this paper. As we are concerned with domains of arbitrary finite size, we will consider the following natural generalization of the DNF representation of constraint relations on the Boolean domain. 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 $O(|D| \cdot k)$. Related previous work has studied restrictions on the SAT problem that lead to tractability [15], 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, here we study a more general representation and do not impose any size restrictions, other than finiteness, on the domain.

The second succinct representation that we study, which we refer to as the *decision diagram* representation, is even more succinct than the GDNF representation. It is based on the well-studied ordered binary decision diagram (OBDD) representation of boolean functions (equivalently, boolean relations), but is generalized so as to permit representation of relations over any finite domain.

We remark that, in related work, Marx [16] studied constraint relations represented by truth tables, a representation that is less concise than the explicit representation.

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

For each of the two succinct representations, we give a complete complexity theoretic classification for structural restriction, which generalises the classification for the bounded arity case obtained in [2, 4]. A structural restriction can be described by a class $\mathcal{A}$ of relational structures; we denote the corresponding restricted succinctly represented CSPs by $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ and $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$. For the GDNF representation, we define the key notion of the *incidence structure* of a relational structure. We then prove that $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is in polynomial time if and only if the incidence structures of the structures in $\mathcal{A}$ *have bounded tree width modulo homomorphic equivalence*; by this, we mean that each such structure could be replaced with a homomorphically equivalent one in such a way that the resulting set of structures has bounded tree width. This result is obtained by taking advantage of the known classification for bounded arity; roughly speaking, we obtain our classification by showing that the problem $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is equivalent in complexity to the CSP over the incidence structures of $\mathcal{A}$, which have bounded arity. Our classification for the decision diagram representation proceeds along similar lines: we define the notion of the *dd-structure* of a relational structure, and prove that $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is in polynomial time if and only if the dd-structures of the structures in $\mathcal{A}$ have bounded tree width modulo homomorphic equivalence. Again, this result is obtained by showing that the problem $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is equivalent in complexity to the CSP over the dd-structures of $\mathcal{A}$, and then leveraging the established results on bounded arity. Note that these results are proved relative to the complexity theoretic assumption FPT $\neq$ W[1],

Constraint language restrictions can also be described by a class $\mathcal{B}$ of structures, and we denote them by $\mathsf{CSP}_{\mathsf{GDNF}}(-, \mathcal{B})$. 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 $\mathsf{CSP}_{\mathsf{GDNF}}(-, \mathcal{B})$ and $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B})$ are in polynomial time if $\mathcal{B}$ is a class of relational structures having a near unanimity polymorphism (Theorem 5.1), or if $\mathcal{B}$ is a class of relational structures invariant under a set function (Theorem 5.2); the corresponding results for explicitly represented constraint relations are from [17, 18].

2. Preliminaries, Definitions, and Basic Facts

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

2.1. Relational structures and homomorphisms. As observed by Feder and Vardi [10], 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 $\mathbf{A}, \mathbf{B}, \ldots$ to denote relational structures, and the corresponding non-boldface letters $A, B, \ldots$ 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.)

To each CSP-instance $I = (V, D, C)$ we can 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 all classes $\mathcal{A}, \mathcal{B}$ of structures we let $\mathsf{CSP}(\mathcal{A}, \mathcal{B})$ be the restricted CSP with instances $(\mathbf{A}, \mathbf{B}) \in \mathcal{A} \times \mathcal{B}$ where $\mathbf{A}$ and $\mathbf{B}$ are specified explicitly. 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 $\mathsf{CSP}(\mathcal{A}, -)$.

2.2. 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 $\mathbf{A}$.

2.3. 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 2.1. *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* $\mathsf{tw}(\mathsf{core}(\mathbf{A})) \leq k$*.*

2.4. 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 2.2 (Dalmau, Kolaitis, and Vardi [2]). *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 2.3 (Grohe [4]). *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.

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* [8].

3. Succinct Representations

In this section, we formally define the classes of succinct problems under study, and make some basic observations concerning them.

3.1. Definitions. For classes $\mathcal{A}$, $\mathcal{B}$ of structures, we define $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, \mathcal{B})$ to be the CSP over all instances $(\mathbf{A}, \mathbf{B}) \in \mathcal{A} \times \mathcal{B}$ where all relations of $\mathbf{B}$ are specified in GDNF. Note that we assume that the relations of $\mathbf{A}$ are given explicitly. We write $\mathsf{CSP}_{\mathsf{GDNF}}(-, \mathcal{B})$ or $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ if $\mathcal{A}$ or $\mathcal{B}$, respectively, is the class of all structures.

Similarly, for classes $\mathcal{A}$, $\mathcal{B}$ of structures, we define $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, \mathcal{B})$ to be the CSP over all instances $(\mathbf{A}, \mathbf{B}) \in \mathcal{A} \times \mathcal{B}$ where all relations of $\mathbf{B}$ are specified by *decision diagrams*, which we define next. As before, we assume that the relations of $\mathbf{A}$ are given explicitly, and use $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B})$ or $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is $\mathcal{A}$ or $\mathcal{B}$, respectively, is the class of all structures.

We now formally define the decision diagram representation. A *decision diagram* is a directed graph $(V_0 \cup \cdots \cup V_k, E)$ where

- the different "layers" V_i are disjoint, that is, $i \neq j$ implies $V_i \cap V_j = \emptyset$,
- $|V_0| = 1$, $|V_k| = 1$,

- E is a multiset containing ordered pairs from $\cup_{i\in[k]}(V_{i-1} \times V_i)$,
- each edge $e \in E$ is labelled via a labelling function $l : E \to D$.

That is, a decision diagram is a "layered" directed graph; the first layer (V_0) and last layer (V_k) each contain a unique vertex. The edges always go from one layer to the following layer.

We say that a tuple $(d_1, \ldots, d_k)$ is *accepted* by the decision diagram $G = (V_0 \cup \cdots \cup V_k, E)$ if there exist vertices $v_1 \in V_1, \ldots, v_{k-1} \in V_{k-1}$ such that $v_0, v_1, \ldots, v_{k-1}, v_k$ is a path in G with $l((v_{i-1}, v_i)) = d_i$ for all $i \in [k]$. (Here, we use v_0 to denote the unique vertex in V_0, and v_k to denote the unique vertex in V_k.) The relation *represented* by the decision diagram $G = (V_0 \cup \cdots \cup V_k, E)$ is the relation containing exactly the tuples $(d_1, \ldots, d_k)$ accepted by G.

Example 3.1. *As an example, consider the relation $E_n = \{(b_1, \ldots, b_n) \in \{0,1\}^n : b_1 \oplus \cdots \oplus b_n = 0\}$, defined for $n \geq 1$. Here, $\oplus$ denotes the exclusive OR (XOR) operation. Each E_n has a decision diagram representation that is of size linear in n. In particular, we can give a decision diagram representation $(V_0 \cup \cdots \cup V_n, E)$ of E_n as follows. Define $V_0 = \{v_0\}$, $V_n = \{v_n\}$, and $V_i = \{v_{i0}, v_{i1}\}$ for $i \in [n-1]$. We define E in the following way. There are two edges coming out of v_0: (v_0, v_{10}) with label 0, and (v_0, v_{11}) with label 1. For $i \in [n-1]$, there are two edges coming out of v_{i0}: $(v_{i0}, v_{(i+1)0})$ with label 0, and $(v_{i0}, v_{(i+1)1})$ with label 1; there are also two edges coming out of v_{i1}: $(v_{i1}, v_{(i+1)0})$ with label 1, and $(v_{i1}, v_{(i+1)1})$ with label 0. Finally, there is one edge coming out of $v_{(n-1)0}$, $(v_{(n-1)0}, v_n)$ with label 0, and one edge coming out of $v_{(n-1)1}$, $(v_{(n-1)1}, v_n)$ with label 1. It is straightforward to verify that for all $i \in [n-1]$, the vertex v_{i0} is reachable if and only if $b_1 \oplus \cdots \oplus b_i = 0$, and the vertex v_{i1} is reachable if and only if $b_1 \oplus \cdots \oplus b_i = 1$. By the definition of the two edges that go from the layer V_{n-1} into v_n, it is clear that the accepted tuples are exactly those such that $b_1 \oplus \cdots \oplus b_n = 0$.*

Remark 3.2. *As the GDNF representation generalizes the representation of Boolean functions by formulas in disjunctive normal form, the decision diagram representation also genearlizes a well known representation of Boolean functions. For Boolean functions, the deterministic version of the decision diagrams (called* ordered binary decision diagrams*) is well studied and of great practical imprortance. We consider the nondeterministic version here, because it is more succinct (exponentially more succinct than the deterministic version), and also exponentially more succinct than the GDNF representation, as we shall see below.*

Let us also remark that decision diagrams may be viewed as nondeterministic finite automata over the alphabet D that only accept words of a fixed length k.

3.2. Relationships and basic facts. We now give some basic relationships among the representations and corresponding CSPs. First, we observe that we may translate from the explicit representation to the GNDF representation, and then from the GDNF representation to the decision diagram representation, as made precise by the following propositions.

Proposition 3.3. *There exists a polynomial-time algorithm that, given an explicitly represented relation, outputs a GDNF representation of the relation.*

Proof. A GDNF-representation of a relation R of arity k is the expression

$$\bigcup_{(b_1,\dots,b_k)\in R} (\{b_1\} \times \dots \times \{b_k\}).$$

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

Proposition 3.4. *There exists a polynomial-time algorithm that, given the GDNF representation of a relation, outputs a decision diagram representation of the relation.*

Proof. Given a GDNF representation $R = \bigcup_{i=1}^{m}(P_{i1} \times \cdots \times P_{ik})$ of a relation R, we can create a decision diagram representation $(V_0 \cup \cdots \cup V_k, E)$ in the following way. We define $V_0 = \{v_0\}$, $V_k = \{v_k\}$, and for all $j \in [k-1]$ we define $V_j = \{v_{1j}, \dots, v_{mj}\}$. Now, we will define the edge set in such a way that for each $i \in [m]$, every tuple in $P_{i1} \times \cdots \times P_{ik}$ is accepted via a path of the form $v_0, v_{i1}, v_{i2}, \dots, v_{i(k-1)}, v_k$. For each $i \in [m]$, we create edges as follows. For each element $d \in P_{i1}$, create an edge from v_0 to v_{i1} with label d. For $j = 2, \dots, k-1$, for each element $d \in P_{ij}$, create an edge from $v_{i(j-1)}$ to v_{ij} with label d. And, for each element $d \in P_{ik}$, create an edge from $v_{i(k-1)}$ to v_k with label d.

Clearly, every tuple of R is accepted by the defined decision diagram. Moreover, the only paths in the created decision diagram from v_0 to v_k are of the form $v_0, v_{i1}, v_{i2}, \dots, v_{i(k-1)}, v_k$ for some $i \in [m]$, and by the definition of the edges it is clear that every accepted tuple is contained in R. □□

The previous propositions imply that (up to polynomial factors) the decision diagram representation is at least as succinct as the GDNF representation, which in turn is at least as succinct as the explicit representation. The GDNF representation is strictly more succinct than the explicit representation in that there is a family of relations, namely $R_n = \{0,1\}^n$ having a GDNF representation of size polynomial (indeed, linear) in the arity, whereas the explicit representation requires exponential size (there are 2^n tuples in R_n). Moreover, the decision diagram representation is strictly more succinct than the GDNF representation: The relations E_n defined in Example 3.1 have linear-size decision diagram representations, but the following proposition shows that any GDNF representation must have exponential size.

Proposition 3.5. *A GDNF representation of E_n must have size at least 2^{n-1}.*

Proof. Consider a GDNF representation $\bigcup_{i=1}^{m}(P_{i1} \times \cdots \times P_{in})$ of E_n. For each $i \in [m]$, we have $(P_{i1} \times \cdots \times P_{in}) \subseteq E_n$. We claim that $|P_{i1}| = \cdots = |P_{in}| = 1$. Suppose that there exists j such that $|P_{ij}| > 1$. Then let $\mathbf{b} = (b_1, \dots, b_n)$ be a tuple in $(P_{i1} \times \cdots \times P_{in})$, and let $\mathbf{b}' = (b'_1, \dots, b'_n)$ be the same tuple but with the jth coordinate changed. We have $\mathbf{b}, \mathbf{b}' \in (P_{i1} \times \cdots \times P_{in})$. However, $b_1 \oplus \cdots \oplus b_n \neq b'_1 \oplus \cdots \oplus b'_n$ and only one of the tuples $\mathbf{b}, \mathbf{b}'$ can be in E_n, contradicting $(P_{i1} \times \cdots \times P_{in}) \subseteq E_n$. Thus, each term $(P_{i1} \times \cdots \times P_{in})$ in the GDNF representation contains only one tuple, and we can lower bound m by the number of tuples in E_n, which is 2^{n-1}. □□

We can now make some simple observations relating the complexity of explicitly and succinctly represented CSPs.

Proposition 3.6. *Let $\mathcal{A}$ and $\mathcal{B}$ be two classes of relational structures.*

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

Proof. (1) and (2) follow from Propositions 3.3 and 3.4, respectively. To prove (3), we argue as follows. By (1) and (2), it suffices to show that $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, \mathcal{B})$ reduces to $\mathsf{CSP}(\mathcal{A}, \mathcal{B})$. Given an instance $(\mathbf{A}, \mathbf{B})$ of $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, \mathcal{B})$, an explicit representation for $\mathbf{B}$ can be computed in a brute-force manner: for each relation R, loop over all tuples, checking to see if the tuple is in R; if so, include it in the explicit representation. Since the arity is bounded, this can be carried out in polynomial time. □□

We may observe that the theorems on structural tractability can be immediately transfered to succinctly represented CSPs in the bounded arity case. This is because, for all classes $\mathcal{A}$ of bounded arity, the problems $\mathsf{CSP}(\mathcal{A}, -)$, $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$, and $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ are polynomial time equivalent (by Proposition 3.6).

Corollary 3.7. *Assume that* FPT $\neq$ W[1]. *Let* $\mathcal{A}$ *be a recursively enumerable class of relational structures of bounded arity. Then* $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ *is in polynomial time if and only if* $\mathcal{A}$ *has bounded tree width modulo homomorphic equivalence. The same statement holds for* $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$.

Proof. Immediate from Proposition 3.6, Theorem 2.2 and Theorem 2.3. □□

Thus, for succinct representations, the case of unbounded arity is really the interesting one.

4. Structural Restrictions

The goal of this section is to establish, for succinctly represented CSPs (of possibly unbounded arity), a characterization of tractable structural restrictions. These characterizations will be analogous to those provided by Theorems 2.2 and 2.3 for explicitly represented CSPs of bounded arity, and indeed will rely on these results.

4.1. GDNF representation. Here, we give our classification of those sets of structures $\mathcal{A}$ such that $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is polynomial-time tractable. We begin by defining the *incidence structure* $\mathsf{inc}(\mathbf{A})$ of a relational structure $\mathbf{A}$; this notion will play a key role in our classification. We will show, roughly speaking, that the set of structures $\mathcal{A}$ in the GDNF representation is equivalent in complexity to the set of structures $\mathsf{inc}(\mathcal{A})$ in the explicit representation. We will then use results on the explicit representation to achieve our classification.

Definition 4.1. *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.

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.

Definition 4.2. *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., [19, 15]). It is easy to see that for every structure $\mathbf{A}$ we have

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

However, the incidence width can be much smaller than the tree width:

Example 4.3. *Let $\mathbf{A}$ be a structure with universe $[n]$ and one n-ary relation $R^{\mathbf{A}}$ that only contains the tuple $(1, \ldots, n)$. Then $\mathsf{tw}(\mathbf{A}) = n - 1$ and $\mathsf{iw}(\mathbf{A}) = 1$.*

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

Before we prove the theorem, let us give an example of a class $\mathcal{A}$ of structures such that $\mathsf{CSP}(\mathcal{A}, -)$ is tractable, but $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is not:

Example 4.5. *Let E be a binary relation symbol, and for every $n \geq 1$, let R_n be an n-ary relation symbol. Let $\mathcal{A}$ be the class of all $\{E, R_n\}$-structures $\mathbf{A} = (A, E^{\mathbf{A}}, R_n^{\mathbf{A}})$, where $(A, E^{\mathbf{A}})$ is an arbitrary graph, $n \geq 1$, $|A| = n$, and $R_n^{\mathbf{A}}$ consists of one tuple $(a_1, \ldots, a_n)$ such that $A = \{a_1, \ldots, a_n\}$. It is easy to see that $\mathsf{CSP}(\mathcal{A}, -)$ is in polynomial time, because to check if a structure $\mathbf{A} \in \mathcal{A}$ as above has a homomorphism to an $\{E, R_n\}$-structure $\mathbf{B}$, one just has to check for all tuples $(b_1, \ldots, b_n) \in \mathbf{B}$ if the mapping $a_i \mapsto b_i$ is a homomorphism.*

It follows from our theorem that $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is not in polynomial time unless FPT $\neq$ W[1], *because clearly the class $\mathcal{A}$ does not have bounded incidence width modulo homomorphic equivalence. Actually, we can prove this intractability result under the weaker assumption that* P $\neq$ NP. *To see this, let $\mathbf{B}_n$ be the $\{E, R_n\}$-structure with universe $B_n = \{red, blue, green\}$,*

$$E^{\mathcal{B}_n} = B_n^2 \setminus \big\{(red, red), (blue, blue), (green, green)\big\},$$

and $R_n^{\mathbf{B}_n} = (B_n)^n$. Let $\mathcal{B} = \{\mathbf{B}_n \mid n \geq 3\}$. Note that $\mathbf{B}_n$ has a GDNF-representation of size $O(n)$. Hence the problem $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, \mathcal{B})$ is polyomial time equivalent to the NP-complete 3-colorability problem.

Remark 4.6. *The structures $\mathbf{A}_n$ of the previous example are* acyclic *structures. It is long known [20] that for classes $\mathcal{A}$ of acyclic structures, the problem* $\mathsf{CSP}(\mathcal{A}, -)$ *is in polynomial time. More recently, this has been generalized to classes $\mathcal{A}$ of* bounded hypertree width *[14], to classes of* bounded generalized hypertree width *[7, 8], and even to classes of* bounded generalized hypertree width modulo homomorphic equivalence *[8]. Let us remark that* generalized hypertree width *is called* cover width *in [8].*

The rest of this section is devoted to a proof of Theorem 4.4. We require some more preparatory lemmas.

Lemma 4.7. *Let $\mathbf{A}$ and $\mathbf{B}$ be relational structures over the same signature.*

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

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

  for all tuples $(R, a_1, \ldots, a_k)$ 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 $\mathbf{B}$.*

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 $(h'(a_1), \ldots, h'(a_k)) = (b_1, \ldots, b_k) \in R^{\mathbf{B}}$. □□

Lemma 4.8. *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.*

Proof. The structures $\mathbf{A}$ and $\mathsf{core}(\mathbf{A})$ are homomorphically equivalent; by use of Lemma 4.7, 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 4.7 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. □□

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 4.9. *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 2.1 and 4.8. □□

Proof.[of Theorem 4.4] The idea of the proof is to give reductions between the problems $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ 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 4.9, $\mathsf{inc}(\mathcal{A})$ has bounded tree width modulo homomorphic equivalence. Thus by Theorem 2.2, $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ is in polynomial time. We show that $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ 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 $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$, 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 $\sigma = \{R_1, \ldots, R_\ell\}$, 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 $1 \le i \le l$, the binary relations $R_{i1}^{\mathbf{B}'}, \ldots, R_{ir_i}^{\mathbf{B}'}$ 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 $(R_i, a_1, \ldots, a_{r_i})$ in the universe of $\mathsf{inc}(\mathbf{A})$, $h'(R_i, a_1, \ldots, a_{r_i})$ is defined to be an element p_{ij} for some $j \in [m_i]$ with $(h(a_i), \ldots, h(a_{r_i})) \in P_{ij1} \times \ldots \times P_{ijr_i}$. Such a j exists, because

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

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 4.4.

For the backward direction, suppose that $\mathcal{A}$ does not have bounded incidence-width modulo homomorphic equivalence. We wish to show that $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$ is not in polynomial time. By Lemma 4.9, $\mathsf{inc}(\mathcal{A})$ 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 2.3. Thus it suffices to give a polynomial-time reduction from $\mathsf{CSP}(\mathsf{inc}(\mathcal{A}), -)$ to $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$. 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 $\mathsf{CSP}_{\mathsf{GDNF}}(\mathcal{A}, -)$. 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. □□

4.2. Decision diagram representation. Here, we give our classification of the polynomial-time tractable problems $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$. Whereas with the GDNF representation, the *incidence structure* of a relational structure played a key role, here a structure that we call the *dd-structure* will similarly play a key role. Indeed, the development in this subsection largely parallels that of the previous subsection.

Definition 4.10. *The* dd-signature $\mathsf{dd}(\sigma)$ *of a relational signature* σ *is a relational signature derived from* σ *in the following way. For every relation symbol* R *of* σ *having arity* 1, R *is also contained in* $\mathsf{dd}(\sigma)$*; and, for every relation symbol* R *of* σ *having arity* $k \geq 2$, *the signature* $\mathsf{dd}(\sigma)$ *contains symbols* $R_1, \ldots, R_k$ *where* R_1 *and* R_k *are of arity* 2, *and* $R_2, \ldots, R_{k-1}$ *are of arity* 3.

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

- *having universe* $A \cup \bigcup_{R \in \sigma} \{y_i^{(R,a_1,\ldots,a_k)} : (a_1, \ldots, a_k) \in R^{\mathbf{A}}, i \in [k-1]\}$, *and*
- *where for each relation symbol* R *of* σ *having arity* 1, *we define*

$$R^{\mathsf{dd}(\mathbf{A})} = R^{\mathbf{A}}$$

and for each relation symbol R *of* σ *having arity* $k \geq 2$, *we define*

$$R_1^{\mathsf{dd}(\mathbf{A})} = \{(a_1, y_1^{(R,a_1,\ldots,a_k)}) : (a_1, \ldots, a_k) \in R^{\mathbf{A}}\}$$

$$R_i^{\mathsf{dd}(\mathbf{A})} = \{(y_{i-1}^{(R,a_1,\ldots,a_k)}, a_i, y_i^{(R,a_1,\ldots,a_k)}) : (a_1, \ldots, a_k) \in R^{\mathbf{A}}\}$$

for $i = 2, \ldots, k-1$

and

$$R_k^{\mathsf{dd}(\mathbf{A})} = \{(y_{k-1}^{(R,a_1,\ldots,a_k)}, a_k) : (a_1, \ldots, a_k) \in R^{\mathbf{A}}\}$$

Definition 4.11. *The* dd-width $\mathsf{ddw}(\mathbf{A})$ *of a relational structure* $\mathbf{A}$ *is the tree width of* $\mathsf{dd}(\mathbf{A})$.

Proposition 4.12. *For all structures* $\mathbf{A}$ *it holds that* $\mathsf{iw}(\mathbf{A}) \leq \mathsf{ddw}(\mathbf{A})$.

Proof. Let $\mathbf{A}$ be a σ-structure, and let (T, Y), where $T = (I, F)$, be a tree decomposition of $\mathsf{dd}(\mathbf{A})$. Observe that for all $R \in \sigma$ and all $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$, the set

$$S = \{t \in I \mid y_i^{(R,a_1,\ldots,a_k)} \in Y_t \text{ for some } i \in [0, k]\}$$

is connected in T, because the set $S_i = \{t \in I \mid y_i^{(R,a_1,\ldots,a_k)} \in Y_t\}$ is connected for every $i \in [0, k]$, and we have $S_{i-1} \cap S_i \neq \emptyset$ for all $i \in [k]$ since

$$(y_{i-1}^{(R,a_1,\ldots,a_k)}, a_i, y_i^{(R,a_1,\ldots,a_k)}) \in R_i^{\mathsf{dd}(A)}.$$

For every $t \in I$ we let

$$X_t := (A \cap Y_t) \cup \{(R, a_1, \ldots, a_k) \mid y_i^{(R,a_1,\ldots,a_k)} \in Y_t \text{ for some } i \in [0,k]\}.$$

Then it is easy to see that (T, X) is a tree decomposition of $\mathsf{inc}(\mathbf{A})$. Obviously, $|X_t| \leq |Y_t|$, hence the width of (T, X) is less than or equal to the width of (T, Y). □□

Example 4.13. *Let $n \geq 1$, and let R be an n^2-ary relation symbol. Let $\mathbf{A}$ be the $\{R\}$-structure with universe $[n]^2$ and*

$$R^{\mathbf{A}} = \big\{(a_1, \ldots, a_{n^2}), (b_1, \ldots, b_{n^2})\big\},$$

where $a_{(i-1)\cdot n+j} = (i, j)$ and $b_{(i-1)\cdot n+j} = (j, i)$ for all $i, j \in [n]$. Hence the a-tuple traverses $[n]^2$, viewed as an $n \times n$-grid, "row-wise" whereas a-tuple traverses $[n]^2$ "column-wise".

Then

$$\mathsf{iw}(\mathbf{A}) = 2 \quad \textit{and} \quad \mathsf{ddw}(\mathbf{A}) \geq n.$$

To see that $\mathsf{iw}(\mathbf{A}) \leq 2$, we construct a tree decomposition (T, X) of $\mathsf{inc}(\mathbf{A})$ as follows: The tree T is a star with one center c and n^2 leaves ℓ_{ij}, for $i, j \in [n]^2$, attached to c. The bag X_c consists of the two elements $(R, a_1, \ldots, a_{n^2}), (R, b_1, \ldots, b_{n^2})$. For every leaf ℓ_{ij}, the bag $X_{\ell_{ij}}$ consists of the three elements

$$(i, j), (R, a_1, \ldots, a_{n^2}), (R, b_1, \ldots, b_{n^2}).$$

The argument that $\mathsf{ddw}(\mathbf{A}) \geq n$ is based on some standard facts from graph theory. The Gaifman graph *of a relational structure is the graph whose vertices are the elements of the structure, with an edge between two vertices if they occur together in some tuple of some relation of the structure. We observe that the Gaifman graph of our structure $\mathsf{dd}(\mathbf{A})$ contains the $n \times n$-grid as a minor. Now we use the facts that (a) a structure has the same tree width as its Gaifman graph, (b) tree width is minor monotone, and (c) the $n \times n$-grid has tree width n (see, for example, [21]; (a) follows immediately from Lemma 12.3.5, (b) is Lemma 12.3.6, and (c) is Exercise 21 of Chapter 12). They imply that $\mathsf{ddw}(\mathbf{A}) = \mathsf{tw}(\mathsf{dd}(\mathbf{A})) \geq n$.*

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

The following is the statement of our classification theorem.

Theorem 4.14. *Assume that* FPT $\neq$ W[1]. *Let $\mathcal{A}$ be a recursively enumerable class of relational structures. Then $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is in polynomial time if and only if $\mathcal{A}$ has bounded dd-width modulo homomorphic equivalence.*

Lemma 4.15. *Let $\mathbf{A}$ and $\mathbf{B}$ be relational structures over the same signature σ.*

- *Let h be a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Then, there is a unique extension h' of h that is a homomorphism from $\mathsf{dd}(\mathbf{A})$ to $\mathsf{dd}(\mathbf{B})$, given by*

  $$h'(y_i^{(R,a_1,\ldots,a_k)}) = y_i^{(R,h(a_1),\ldots,h(a_k))}$$

  for $i \in [k-1]$ and over all tuples $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$ and all $R \in \sigma$.

- *The restriction to A of any homomorphism from* $\mathsf{dd}(\mathbf{A})$ *to* $\mathsf{dd}(\mathbf{B})$ *is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$.

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 $\mathsf{dd}(\mathbf{A})$ to $\mathsf{dd}(\mathbf{B})$ extending h. Let $y_i^{(R,a_1,\ldots,a_k)}$ be an element from the universe of $\mathsf{dd}(\mathbf{A})$. We claim that $g(y_i^{(R,a_1,\ldots,a_k)}) = y_i^{(R,h(a_1),\ldots,h(a_k))}$. First, observe that $(a_1, y_1^{(R,a_1,\ldots,a_k)}) \in R_1^{\mathsf{dd}(\mathbf{A})}$, implying that $(g(a_1), g(y_1^{(R,a_1,\ldots,a_k)})) \in R_1^{\mathsf{dd}(\mathbf{B})}$. By definition of $\mathsf{dd}(\mathbf{B})$, we have that $g(y_i^{(R,a_1,\ldots,a_k)})$ is a tuple of the form $y_i^{(R,b_1,\ldots,b_k)}$. By the definition of $\mathsf{dd}(\mathbf{B})$, we have $g(a_1) = b_1$ and thus we have $h(a_1) = b_1$. We prove by induction that for $i > 1$ it holds that $g(y_i^{(R,a_1,\ldots,a_k)}) = y_i^{(R,b_1,\ldots,b_k)}$ and $g(a_i) = b_i$ (and thus we have $h(a_i) = b_i$). We have, by definition of $\mathsf{dd}(\mathbf{A})$, that (when $i < k$) $(y_{i-1}^{(R,a_1,\ldots,a_k)}, a_i, y_i^{(R,a_1,\ldots,a_k)}) \in R_i^{\mathsf{dd}(\mathbf{A})}$. Since by induction we have $g(y_{i-1}^{(R,a_1,\ldots,a_k)}) = y_{i-1}^{(R,b_1,\ldots,b_k)}$, by the definition of $R_i^{\mathsf{dd}(\mathbf{B})}$ we have $g(a_i) = b_i$ and $g(y_i^{(R,a_1,\ldots,a_k)}) = y_i^{(R,b_1,\ldots,b_k)}$. The reasoning for $i = k$ is similar.

For the second claim, by the argument just given, for any homomorphism g from $\mathsf{dd}(\mathbf{A})$ to $\mathsf{dd}(\mathbf{B})$, we have that for any tuple $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$ that there exists a tuple $(b_1, \ldots, b_k) \in B^k$ such that $g(y_i^{(R,a_1,\ldots,a_k)}) = y_i^{(R,b_1,\ldots,b_k)}$ for all $i \in [k-1]$, and that $g(a_i) = b_i$ for all $i \in [k]$. By the definition of the $R_i^{\mathsf{dd}(\mathbf{B})}$, we have $(b_1, \ldots, b_k) \in R^{\mathbf{B}}$. □□

Lemma 4.16. *For any relational structure* $\mathbf{A}$*, it holds that the relational structures* $\mathsf{core}(\mathsf{dd}(\mathbf{A}))$ *and* $\mathsf{dd}(\mathsf{core}(\mathbf{A}))$ *are isomorphic.*

Proof. The structure of this proof is identical to that of Lemma 4.8. The structures $\mathbf{A}$ and $\mathsf{core}(\mathbf{A})$ are homomorphically equivalent; by Lemma 4.15, it follows that the structures $\mathsf{dd}(\mathbf{A})$ and $\mathsf{dd}(\mathsf{core}(\mathbf{A}))$ are homomorphically equivalent. To establish the desired isomorphism, it suffices to show that the structure $\mathsf{dd}(\mathsf{core}(\mathbf{A}))$ is a core.

Let $\mathbf{C}$ be a core of $\mathbf{A}$. Let $h' : \mathsf{dd}(\mathbf{C}) \to \mathsf{dd}(\mathbf{C})$ 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 tuple $(c_1, \ldots, c_k) \in R^{\mathbf{C}}$ there exists another tuple $(b_1, \ldots, b_k) \in R^{\mathbf{C}}$ such that $(h(b_1), \ldots, h(b_k)) = (c_1, \ldots, c_k)$. Now let $y_i^{(R,c_1,\ldots,c_k)}$ be an arbitrary element that is in the universe of $\mathsf{dd}(\mathbf{C})$, but outside of C. By definition, it holds that $(c_1, \ldots, c_k) \in R^{\mathbf{C}}$ and as we just showed, there exists $(b_1, \ldots, b_k) \in R^{\mathbf{C}}$ such that $(h(b_1), \ldots, h(b_k)) = (c_1, \ldots, c_k)$. By Lemma 4.15, we have that $h(y_i^{(R,b_1,\ldots,b_k)}) = y_i^{(R,c_1,\ldots,c_k)}$. We conclude that h' is surjective. □□

Lemma 4.17. *For every class $\mathcal{A}$ of relational structures, the following four statements are equivalent:*

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

Proof. Follows immediately from Lemmas 4.15 and 4.16. □□

Proof. (Theorem 4.14) We give reductions between the problems $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ and $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$. First, we assume that $\mathcal{A}$ has bounded dd-width modulo homomorphic equivalence. Then by Lemma 4.17, $\mathsf{dd}(\mathcal{A})$ has bounded tree width modulo homomorphic equivalence. Thus, by Theorem 2.2, $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$ is in polynomial time. We show that $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is in polynomial time by giving a polynomial-time reduction to $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$.

We reduce an instance $(\mathbf{A}, \mathbf{B})$ of $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$, where the relations of $\mathbf{B}$ are represented by decision diagrams, to an instance $(\mathsf{dd}(\mathbf{A}), \mathbf{B}')$ of $\mathsf{CSP}(\mathsf{dd}(\mathbf{A}), -)$ for a relational structure $\mathbf{B}'$ defined as follows. For each relation symbol R, denote the decision diagram for $R^{\mathbf{B}}$ as $(V_0 \cup \cdots \cup V_k, E)$ and define

$$R_1^{\mathbf{B}'} = \{(d, v_1) : v_1 \in V_1, (v_0, v_1) \in E, d = l((v_0, v_1))\}$$

$$R_i^{\mathbf{B}'} = \{(v_{i-1}, d, v_i) : v_{i-1} \in V_{i-1}, v_i \in V_i, (v_{i-1}, v_i) \in E, d = l((v_{i-1}, v_i))\}$$

$$R_k^{\mathbf{B}'} = \{(v_{k-1}, d) : v_{k-1} \in V_{k-1}, (v_{k-1}, v_k) \in E, d = l((v_{k-1}, v_k))\}$$

Here, v_0 and v_k denote the unique elements of V_0 and V_k, respectively. It is clear that $\mathbf{B}'$ can be constructed from the succinct representation of $\mathbf{B}$ in polynomial time. Note that the universe of $\mathbf{B}'$ is the union of the sets

$$B \cup V_1 \cup \cdots \cup V_{k-1}$$

over all decision diagrams for relations $R^{\mathbf{B}}$ of $\mathbf{B}$.

Suppose that there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$. We may extend h to a homomorphism from $\mathsf{dd}(\mathbf{A})$ to $\mathbf{B}'$ as follows. For each tuple $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$, select vertices $v_1, \ldots, v_{k-1}$ witnessing that $(h(a_1), \ldots, h(a_k)) \in R^{\mathbf{B}}$, and define $h'(y_i^{(R, a_1, \ldots, a_k)}) = v_i$ for $i \in [k-1]$. It follows from the definition of $\mathbf{B}'$ that this is a homomorphism.

Now suppose that there is a homomorphism h from $\mathsf{dd}(\mathbf{A})$ to $\mathbf{B}'$. We claim that $h|_A$ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Let $(a_1, \ldots, a_k)$ be a tuple in $R^{\mathbf{A}}$. We have that $(h(a_1), \ldots, h(a_k))$ is accepted by the decision diagram for $R^{\mathbf{B}}$ via the vertices $h(y_1^{(R, a_1, \ldots, a_k)}), \ldots, h(y_{k-1}^{(R, a_1, \ldots, a_k)})$.

For the other direction of the theorem, suppose that $\mathcal{A}$ does not have bounded dd-width modulo homomorphic equivalence. We want to show that $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$ is not in polynomial time. By Lemma 4.17, $\mathsf{dd}(\mathcal{A})$ does not have bounded tree width modulo homomorphic equivalence. Noting that the recursive enumerability of $\mathcal{A}$ implies the recursive enumerability of $\mathsf{dd}(\mathcal{A})$ and that $\mathsf{dd}(\mathcal{A})$ has arity bounded by 3, $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$ is not in polynomial time by Theorem 2.3. We give a polynomial-time reduction from $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$ to $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$. Given an instance $(\mathsf{dd}(\mathbf{A}), \mathbf{B})$ of $\mathsf{CSP}(\mathsf{dd}(\mathcal{A}), -)$, we demonstrate how to create an instance $(\mathbf{A}, \mathbf{B}')$ of $\mathsf{CSP}_{\mathsf{DD}}(\mathcal{A}, -)$.

Let R be a relation symbol. For all $i \in [k-1]$, define $V_i = \{d[i] : d \in \pi_1(R_{i+1})\}$ where π_1 denotes the projection onto the first coordinate. Note that the bracket notation is added to ensure that the sets V_i are disjoint. Let v_0, v_k be symbols not appearing in any of the sets $V_1, \ldots, V_{k-1}$ and define $V_0 = \{v_0\}$, $V_k = \{v_k\}$. Let

$$E_1 = \{(v_0, b, d[1]) : (b, d) \in R_1^{\mathbf{B}}\}$$

$$E_i = \{(d[i-1], b, d'[i]) : (d, b, d') \in R_i^{\mathbf{B}}\}$$

for $i = 2, \ldots, k-1$

$$E_k = \{(d[k-1], b, v_k) : (d, b) \in R_k\}$$

We define $E = E_1 \cup \cdots \cup E_k$, and define the decision diagram for $R^{\mathbf{B}'}$ to be $(V_0 \cup \cdots \cup V_k, E)$.

Suppose there is a homomorphism h from $\mathsf{dd}(\mathbf{A})$ to $\mathbf{B}$. We claim that $h|_A$ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}'$. Consider a tuple $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$. By definition of $\mathsf{dd}(\mathbf{A})$, we have

$$(h(a_1), h(y_1^{(R,a_1,\ldots,a_k)})) \in R_1^{\mathbf{B}}$$

$$(h(y_{i-1}), h(a_i), h(y_i)) \in R_i^{\mathbf{B}}$$

for $i = 2, \ldots, k-1$

$$(h(y_{k-1}), h(a_k)) \in R_k^{\mathbf{B}}$$

By definition of $\mathbf{B}'$, we have that the tuple $(h(a_1), \ldots, h(a_k))$ is accepted by $R^{\mathbf{B}'}$ via the vertices $(h(y_1^{(R,a_1,\ldots,a_k)}))[1], \ldots, (h(y_{k-1}^{(R,a_1,\ldots,a_k)}))[k-1]$.

Now suppose that there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}'$. We show that it can be extended to a homomorphism h' from $\mathsf{dd}(\mathbf{A})$ to $\mathbf{B}$. Let $(a_1, \ldots, a_k) \in R^{\mathbf{A}}$ be a tuple of $\mathbf{A}$. We have $(h(a_1), \ldots, h(a_k))$ is accepted by $R^{\mathbf{B}}$; let $d_1[1], \ldots, d_{k-1}[k-1]$ be vertices witnessing this. Define $h'(y_i^{(R,a_1,\ldots,a_k)}) = d_i$ for all $i \in [k-1]$. It follows from the definition of $\mathbf{B}'$ that h' is a homomorphism from $\mathsf{dd}(\mathbf{A})$ to $\mathbf{B}$. □□

5. Constraint language restrictions

This section presents a pair of tractability results based on constraint language restrictions. The first result is based on *near-unanimity polymorphisms* which were studied in the CSP (with explicitly represented tuples) in [17]. 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 5.1. *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 problems $\mathsf{CSP}_{\mathsf{GDNF}}(-, \mathcal{B}_k)$ and $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B}_k)$ are in polynomial time.*

Proof. By Proposition 3.6, it suffices to prove the tractability result for $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B}_k)$. We give a reduction from $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B}_k)$ to $\mathsf{CSP}(-, \mathcal{B}_k)$; the latter is tractable by the "strong k-consistency" algorithm, see [17] for a description of this algorithm and proof. Note that this algorithm, for any fixed k, runs in polynomial time. Given an instance ϕ of $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B}_k)$, we create an instance ϕ' of $\mathsf{CSP}(-, \mathcal{B}_k)$ containing all $(k-1)$-*projections* of constraints in ϕ. A $(k-1)$-projection of a constraint $Rx_1 \ldots x_n$ is a constraint $Sx_{i_1} \ldots x_{i_{k-1}}$ where $i_1, \ldots, i_{k-1}$ is a subsequence of $1, \ldots, n$, and $S = \{(b_{i_1}, \ldots, b_{i_{k-1}}) :$

$(b_1, \ldots, b_n) \in R\}$. By the $(k-1)$-decomposability of constraints having a near-unanimity operation of arity k [17], ϕ' is satisfiable if and only if ϕ is.

For ease of notation, we describe the reduction using the definition of CSP given in the introduction. For each constraint $Rx_1 \ldots x_n$ in ϕ, we create constraints in ϕ' as follows. If $n < (k-1)$, we simply create a copy of $Rx_1 \ldots x_n$ in ϕ', but where R is represented explicitly; we may create the explicit representation from the decision diagram representation as in Proposition 3.6. Otherwise, for each subsequence $i_1, \ldots, i_{k-1}$ of $1, \ldots, n$, we compute the projection $Sx_{i_1} \ldots x_{i_{k-1}}$ of $Rx_1 \ldots x_n$ onto $x_{i_1}, \ldots, x_{i_{k-1}}$ as follows. For each tuple $b_1, \ldots, b_{k-1}$, we can determine if $(b_1, \ldots, b_{k-1}) \in S$ as follows. From the decision diagram representation of R, for $j = 1, \ldots, k-1$, eliminate all edges in $V_{i_j} \times V_{i_j+1}$ that do not have label b_j. Then, there is a path from $v_0 \in V_0$ to $v_k \in V_k$ if and only if $(b_1, \ldots, b_{k-1}) \in S$. By performing this procedure for every tuple, we may compute S. □□

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 [18].

Theorem 5.2. *Let $\mathcal{B}$ be the set of all succinctly specified relational structures invariant under a set function. The problems $\mathsf{CSP}_{\mathsf{GDNF}}(-, \mathcal{B})$ and $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B})$ are in polynomial time.*

Proof. By Proposition 3.6, it suffices to prove the tractability result for $\mathsf{CSP}_{\mathsf{DD}}(-, \mathcal{B})$. We show how to implement the *arc consistency* procedure on succinctly represented instance. This is a well known procedure for CSPs that is known to solve, in polynomial time, explicitly represented CSPs that are invariant under a set function; see the papers [18, 22] for more information on arc consistency and tractability.

It suffices to show that we can perform, in polynomial time, two operations on constraints represented using decision diagrams. First, computing a projection onto one coordinate, that is, from a constraint $Rx_1 \ldots x_n$, computing the set $\{b_i : (b_1, \ldots, b_n) \in R\}$ for any $i \in [n]$. Second, for a subset S of the domain and a coordinate $i \in [n]$, computing the relation $R' = \{(b_1, \ldots, b_n) \in R : b_i \in S\}$.

For the first operation, we can proceed as in the proof of Theorem 5.1: for each domain value b, remove all edges not having b as label from $V_{i-1} \times V_i$, and then check to see if $v_k \in V_k$ is reachable from $v_0 \in V_0$. The domain values such that this check is successful is the desired subset. For the second operation, we may simply remove all edges not having a label in S from $V_{i-1} \times V_i$. □□

6. 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. We obtain surprisingly simple and clean-cut characterizations of tractable structural restrictions for the two representations studied. No corresponding classification result is known for explicitly represented CSPs.

While we obtain complete classifications for the structural restrictions of succinctly specified CSPs, we are far from such a classification for constraint language restrictions. This is not very surprising, because constraint language restrictions are also “more difficult” for explicitly represented CSPs. Nevertheless, it might be possible to obtain further tractability results for succinctly represented CSPs with a restricted constraint language. It would be interesting to gain a better understanding of how the algebraic approach to classifying constraint language restrictions interacts with succinct representations.

There might be further interesting succinct representations besides the two studied here. In particular, it might be worthwhile to study deterministic decision diagrams. Obviously, they are less succinct than (nondeterministic) decision diagrams, and it can easily be seen that they are incomparable to the GDNF representation. Of course over specific domains there are other natural representations of the constraint relations such as linear (in)equalities, polynomial (in)equalities, et cetera. But these represenations seem to have a different flavor than our “general purpose” succinct representations.

[1] A. Bulatov, A dichotomy theorem for constraints on a three-element set, in: Proceedings of 43rd IEEE Symposium on Foundations of Computer Science, 2002, pp. 649–658.

[2] V. Dalmau, P. G. Kolaitis, M. Y. Vardi, Constraint satisfaction, bounded treewidth, and finite-variable logics, in: Constraint Programming ’02, LNCS, 2002, pp. 310–326.

[3] A. Bulatov, Tractable conservative constraint satisfaction problems, in: Proceedings of 18th IEEE Symposium on Logic in Computer Science (LICS ’03), 2003, pp. 321–330, extended version appears as Oxford University technical report PRG-RR–03-01.

[4] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, Journal of the ACM 54 (2007) 1–24.

[5] V. Dalmau, Mal’tsev constraints made simple, eCCC technical report (2004).

[6] D. Cohen, P. Jeavons, M. Gyssens, A unified theory of structural tractability for constraint satisfaction and spread cut decomposition, iJCAI 2005 (2005).

[7] I. Adler, G. Gottlob, M. Grohe, Hypertree-width and related hypergraph invariants, in: S. Felsner (Ed.), Proceedings of the 3rd European Conference on Combinatorics, Graph Theory, and Applications, Vol. AE of DMTCS Proceedings Series, 2005, pp. 5–10.

[8] H. Chen, V. Dalmau, Beyond hypertree width: Decomposition methods without decompositions, in: P. van Beek (Ed.), Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming, Vol. 3709 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 167–181.

[9] M. Grohe, D. Marx, Constraint solving via fractional edge covers, in: Proceedings of the of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006), 2006, pp. 289–298.

[10] T. Feder, M. Y. Vardi, The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory, SIAM J. Comput. 28 (1) (1998) 57–104.

[11] R. Dechter, J. Pearl, Tree clustering for constraint networks, Artificial Intelligence (1989) 353–366.

[12] E. Freuder, Complexity of k-tree structured constraint satisfaction problems, in: AAAI-90, 1990, pp. 4–9.

[13] G. Gottlob, N. Leone, F. Scarcello, A comparison of structural csp decomposition methods, Artif. Intell. 124 (2) (2000) 243–282.

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

[15] S. 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 (2004) 188–202.

[16] D. Marx, Tractable structures for constraint satisfaction with truth tables, in: S. Albers, J.-Y. Marion (Eds.), 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009), Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany, 2009, pp. 649–660.
URL `http://drops.dagstuhl.de/opus/volltexte/2009/1807`

[17] P. Jeavons, D. Cohen, M. Cooper, Constraints, consistency, and closure, Articial Intelligence 101 (1-2) (1998) 251–265.
[18] V. Dalmau, J. Pearson, Closure functions and width 1 problems, in: CP 1999, 1999, pp. 159–173.
[19] J. Flum, M. Frick, M. Grohe, Query evaluation via tree-decompositions, J. ACM 49 (6) (2002) 716–752.
[20] M. Yannakakis, Algorithms for acyclic database schemes, in: 7th International Conference on Very Large Data Bases, 1981, pp. 82–94.
[21] R. Diestel, Graph Theory, 3rd Edition, Springer Verlag, 2005.
[22] H. Chen, V. 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.