Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics

Víctor Dalmau[1], Phokion G. Kolaitis ⋆[2], and Moshe Y. Vardi ⋆⋆[3]

[1] Universitat Pompeu Fabra, Barcelona
[2] University of California, Santa Cruz
[3] Rice University

Abstract. We systematically investigate the connections between constraint satisfaction problems, structures of bounded treewidth, and definability in logics with a finite number of variables. We first show that constraint satisfaction problems on inputs of treewidth less than k are definable using Datalog programs with at most k variables; this provides a new explanation for the tractability of these classes of problems. After this, we investigate constraint satisfaction on inputs that are homomorphically equivalent to structures of bounded treewidth. We show that these problems are solvable in polynomial time by establishing that they are actually definable in Datalog; moreover, we obtain a logical characterization of the property "being homomorphically equivalent to a structure of bounded treewidth" in terms of definability in finite-variable logics. Unfortunately, this expansion of the tractability landscape comes at a price, because we also show that, for each $k \geq 2$, determining whether a structure is homomorphically equivalent to a structure of treewidth less than k is an NP-complete problem. In contrast, it is well known that, for each $k \geq 2$, there is a polynomial-time algorithm for testing whether a given structure is of treewidth less than k. Finally, we obtain a logical characterization of the property "having bounded treewidth" that sheds light on the complexity-theoretic difference between this property and the property 'being homomorphically equivalent to a structure of bounded treewidth".

1 Introduction and Summary of Results

Constraint satisfaction problems are ubiquitous in several different areas of artificial intelligence, computer science, algebra, logic, and combinatorics. An instance of a constraint-satisfaction problem consists of a set of variables, a set of possible values, and a set of constraints on tuples of variables; the question is to determine whether there is an assignment of values to the variables that satisfies the given constraints. A particularly fruitful way to formalize the above informal description, articulated first by Feder and Vardi [FV98], is to identify the CONSTRAINT SATISFACTION PROBLEM with the HOMOMORPHISM PROBLEM: given two relational structures $\mathbf{A}$ and $\mathbf{B}$, is there a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$? Intuitively, the structure $\mathbf{A}$ represents the variables and

⋆ Supported in part by NSF grant IIS-9907419.
⋆⋆ Supported in part by NSF grants CCR-9988322, IIS-9908435, IIS-9978135, and EIA-0086264, and by BSF grant 9800096.

the tuples of variables that participate in constraints, the structure $\mathbf{B}$ represents the domain of values and the tuples of values that these constrained tuples of variables are allowed to take, and the homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ are precisely the assignments of values to variables that satisfy the constraints. For instance, 3-COLORABILITY is equivalent to the problem of deciding whether there is a homomorphism h from a given graph $\mathbf{G}$ to $\mathbf{K}_3$, where $\mathbf{K}_3$ is the complete graph with 3 nodes. This identification makes it possible to approach constraint satisfaction problems from an algebraic perspective [Jea98,FV98]. Moreover, it makes transparent the connection between constraint satisfaction problems and certain fundamental problems in database theory, such as conjunctive query evaluation and conjunctive query containment (or implication) [KV00a].

Since in its full generality the HOMOMORPHISM PROBLEM is NP-complete [GJ79], researchers have intensively pursued tractable cases of this problem, often referred to as "islands of tractability", that are obtained by imposing restrictions on the input structures $\mathbf{A}$ and $\mathbf{B}$. If σ is a relational vocabulary, and $\mathcal{A}$, $\mathcal{B}$ are two classes of finite σ-structures, then $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$ is the following decision problem: given a structure $\mathbf{A} \in \mathcal{A}$ and a structure $\mathbf{B} \in \mathcal{B}$, is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$? In other words, $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$ is the restriction of the HOMOMORPHISM PROBLEM to inputs from $\mathcal{A}$ and $\mathcal{B}$. If the class $\mathcal{B}$ consists of a single structure $\mathbf{B}$, then we write $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ instead of $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$. Furthermore, if $\mathcal{A}$ is the class $\mathcal{F}(\sigma)$ of all finite σ-structures, then we simply write $\mathrm{CSP}(\mathbf{B})$ instead of $\mathrm{CSP}(\mathcal{F}(\sigma), \mathbf{B})$.

Note that if $\mathcal{G}$ is the class of all undirected graphs and $\mathbf{K}_3$ is the complete graph with 3 nodes, then $\mathrm{CSP}(\mathcal{G}, \mathbf{K}_3)$ is the 3-COLORABILITY problem. Consequently, there are fixed structures $\mathbf{B}$ such that constraint satisfaction problems of the form $\mathrm{CSP}(\mathbf{B})$ are NP-complete. It is a major open problem to characterize those structures $\mathbf{B}$ for which $\mathrm{CSP}(\mathbf{B})$ is tractable. Closely related to this problem is the Feder-Vardi [FV98] Dichotomy Conjecture, which asserts that for every fixed structure $\mathbf{B}$ either $\mathrm{CSP}(\mathbf{B})$ is NP-complete or $\mathrm{CSP}(\mathbf{B})$ is solvable in polynomial time. Although special cases of this conjecture have been confirmed (see, for instance, [Sch78,HN90]), the full conjecture has not been settled thus far. Nonetheless, research on these open problems has led to the discovery of numerous tractable cases of constraint satisfaction (see [Jea98]).

Feder and Vardi [FV98] identified two general sufficient conditions for tractability of $\mathrm{CSP}(\mathbf{B})$ that are broad enough to account for essentially all tractable cases of $\mathrm{CSP}(\mathbf{B})$ that were known at that time. One of these two conditions is group-theoretic, while the other has to do with expressibility of constraint satisfaction problems in Datalog, the main query language for deductive databases [Ull89]. More precisely, Feder and Vardi [FV98] showed that for many polynomial-time solvable constraint satisfaction problems of the form $\mathrm{CSP}(\mathbf{B})$ there is a Datalog program that defines the complement $\neg\mathrm{CSP}(\mathbf{B})$ of $\mathrm{CSP}(\mathbf{B})$.

Tractable constraint satisfaction problems of the form $\mathrm{CSP}(\mathbf{B})$ represent restricted cases of the HOMOMORPHISM PROBLEM "is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$?" in which $\mathbf{B}$ is kept fixed and also required to satisfy certain additional conditions that imply tractability. Tractable cases of constraint satisfaction can also be obtained, however, by imposing conditions on $\mathbf{A}$ while letting $\mathbf{B}$ be arbitrary. In particular, an important large "island of tractability" is formed by the class of structures of *bounded treewidth*, where the *treewidth* of a relational structure is a positive integer that measures how “close” to a tree the structure is. Specifically, Dechter and Pearl [DP89] and Freuder [Fre90] have shown that, for every $k \geq 2$, the constraint satisfaction problem $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ is solvable in polynomial time, where $\mathcal{T}^k(\sigma)$ is the class of all σ-structures of treewidth less than k (and, as before, $\mathcal{F}(\sigma)$ is the class of all σ-structures). In [KV00b], a different proof of this result was obtained by exploiting the tight connection between the constraint satisfaction and conjunctive query evaluation, as well as the tractability of query evaluation for fragments of first-order logic with a finite number of variables. If $\mathbf{A}$ is a relational structure, then the *canonical conjunctive query* of $Q^{\mathbf{A}}$ is a positive existential first-order sentence that describes which tuples from $\mathbf{A}$ are in the various relations of $\mathbf{A}$. Chandra and Merlin [CM77] pointed out that, given two structures $\mathbf{A}$ and $\mathbf{B}$, a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ exists if and only if $\mathbf{B}$ satisfies $Q^{\mathbf{A}}$. In general, $Q^{\mathbf{A}}$ requires as many variables as elements in the universe of $\mathbf{A}$. In [KV00b], however, it was shown that if $\mathbf{A}$ is of treewidth less than k, then k variables suffice to express $Q^{\mathbf{A}}$, i.e., $Q^{\mathbf{A}}$ is equivalent to a sentence of L^k, which is the fragment of first-order logic with k variables containing all atomic formulas in these k variables and closed only under conjunction and existential quantification over these variables. The tractability of $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ follows then by combining this result with the fact that the evaluation problems for L^k-sentences is polynomial-time solvable, which follows from more general results in [Var95].

Our goal in this paper is to systematically explore the connections between constraint satisfaction problems, structures of bounded treewidth, and definability in logics with a finite number of variables. The first main result asserts that definability in Datalog provides also an explanation for the tractability of constraint satisfaction problems on structures of bounded treewidth. Specifically, we show that, for every $k \geq 2$ and every σ-structure $\mathbf{B}$, the complement $\neg\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathbf{B})$ is expressible in k-Datalog, i.e., it is definable by a Datalog program with k variables in the body and the head of each rule. From this it follows that, for every $k \geq 2$, $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ is definable in LFP^{2k}, where LFP^{2k} is the fragment of least fixed-point logic with $2k$ variables. Since query evaluation in this fragment is solvable in polynomial time, this result provides another proof of the tractability of the constraint satisfaction problem on structures of bounded treewidth. We also show that testing whether strong k-consistency can be established on $\mathbf{A}$ and $\mathbf{B}$ is a sound and complete algorithm for determining whether there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, when $\mathbf{A}$ is a structure of treewidth less than k. The proofs of these results make use of certain connections between constraint satisfaction, finite-variable logics, and combinatorial pebble games that were studied in [KV00b].

After this, we turn attention on the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$, of all σ-structures $\mathbf{A}$ that are *homomorpically equivalent* to a σ-structure $\mathbf{D}$ of treewidth less than k (i.e., there are homorphisms from $\mathbf{A}$ to $\mathbf{D}$ and from $\mathbf{D}$ to $\mathbf{A}$). Clearly, each of these classes properly contains the class $\mathcal{T}^k(\sigma)$. We show that $\mathrm{CSP}(\mathcal{H}(\mathcal{T}^k(\sigma)), \mathcal{F}(\sigma))$ is solvable in polynomial time by establishing that $\neg\mathrm{CSP}(\mathcal{H}(\mathcal{T}^k(\sigma)), \mathbf{B})$ is in k-Datalog, for every σ-structure $\mathbf{B}$. Thus, the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$, constitute new large “islands of tractability” for constraint satisfaction and, moreover, their tractability is once again due to definability in Datalog. We then proceed to characterize $\mathcal{H}(\mathcal{T}^k(\sigma))$ in terms of definability in finite-variable logics by showing that for each $k \geq 2$, a structure $\mathbf{A}$ is homomorphically equivalent to a structure of treewidth less than k if and only if the canonical query $Q^{\mathbf{A}}$ of $\mathbf{A}$ is logically equivalent to an L^k-sentence.

The above properties of the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$, appear to make them large and appealing "islands of tractability". Unfortunately, this expansion of the tractability landscape comes at a price, because accessing these new "islands of tractability" turns out to be a hard problem. Indeed, we show that, for every $k \geq 2$, testing for membership in $\mathcal{H}(\mathcal{T}^k(\sigma))$ is an NP-complete problem. This should be contrasted with the state of affairs for $\mathcal{T}^k$, since it is well known that, for every $k \geq 2$, testing for membership in $\mathcal{T}^k$ is solvable in polynomial time [Bod93].

Our study of the connections between bounded treewidth and finite-variable logics culminates with a logical characterization of bounded treewidth that sheds light on the differences between $\mathcal{T}^k(\sigma)$ and $\mathcal{H}(\mathcal{T}^k(\sigma))$. For this, we analyze a set of *rewriting rules* that are widely used in database query processing and show that, for each $k \geq 2$, a structure $\mathbf{A}$ has treewidth less than k if and only if the canonical query $Q^{\mathbf{A}}$ of $\mathbf{A}$ can be rewritten to an L^k-sentence using these rules.

2 Preliminaries and Background

A *vocabulary* σ is a finite set $\{R_1, \ldots, R_m\}$ of relation symbols of specified arities. A σ*-structure* is a relational structure of the form $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$, where each $R_i^{\mathbf{A}}$ is a relation on the universe A of $\mathbf{A}$ such that the arity of $R_i^{\mathbf{A}}$ matches that of the relation symbol R_i. We write $\mathcal{F}(\sigma)$ for the class of all finite σ-structures, i.e., σ-structures with a finite set as universe. In what follows, we will assume that all structures under consideration are finite; for this reason, the term "σ-structure" should be understood to mean "finite σ-structure" (on a few occasions, however, we will spell out "finite σ-structure" for emphasis). Also, whenever we refer to *undirected graphs* we mean structures of the form $\mathbf{G} = (V, E)$ such that E is a symmetric binary relation on V without self-loops, i.e., E contains no pairs of the form (v, v), where $v \in V$.

2.1 Conjunctive Queries and Homomorphisms

An n-ary *conjunctive query* Q over a vocabulary σ is a query definable by a positive existential first-order formula over σ having conjunction as its only propositional connective, i.e., by a formula of the form $(\exists z_1) \cdots (\exists z_s)\psi(x_1, \ldots, x_n, z_1, \ldots, z_s)$, where $\psi(x_1, \ldots, x_n, z_1, \ldots, z_s)$ is a conjunction of atomic formulas over σ. For example, the binary conjunctive query "there is a path of length 3 from x_1 to x_2" is definable by the formula $(\exists z_1)(\exists z_2)(E(x_1, z_1) \wedge E(z_1, z_2) \wedge E(z_2, x_2))$. A *Boolean conjunctive query* is definable by a positive existential first-order sentence having conjunction as its only propositional connective, i.e., all variables of ψ have been quantified out.

Every finite σ-structure $\mathbf{A}$ gives rise to a *canonical* Boolean conjunctive query $Q^{\mathbf{A}}$; the positive existential first-order sentence defining $Q^{\mathbf{A}}$ asserts that there exist as many elements as the cardinality of the universe of $\mathbf{A}$ and states all atomic facts satisfied by tuples from the universe of $\mathbf{A}$. For example, if $\mathbf{A} = (A, E)$ is the graph with $A = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$, then $Q^{\mathbf{A}}$ is definable by the sentence

$$(\exists x_1)(\exists x_2)(\exists x_3)(\exists x_4)(E(x_1, x_2) \wedge E(x_2, x_3) \wedge E(x_3, x_4) \wedge E(x_4, x_1)).$$

In what follows, we will mildly abuse the notation by using $Q^{\mathbf{A}}$ to denote both the canonical conjunctive query $Q^{\mathbf{A}}$ associated with the structure $\mathbf{A}$ and the positive existential first-order sentence that defines the canonical query $Q^{\mathbf{A}}$.

If $\mathbf{A}$ and $\mathbf{B}$ are σ-structures, then a *homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$ is a mapping $h : A \mapsto B$ from the universe A of $\mathbf{A}$ to the universe B of $\mathbf{B}$ such that for every relation symbol R_i of σ and every tuple $(a_1, \ldots, a_n) \in R_i^{\mathbf{A}}$, we have that $(h(a_1), \ldots, h(a_n)) \in R_i^{\mathbf{B}}$. Chandra and Merlin discovered the following fundamental result.

Theorem 1. [CM77] *The following are equivalent for finite σ-structures $\mathbf{A}$ and $\mathbf{B}$.*

1. *There is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$.*
2. *$Q^{\mathbf{B}} \models Q^{\mathbf{A}}$, i.e., $Q^{\mathbf{B}}$ logically implies $Q^{\mathbf{A}}$.*
3. *$\mathbf{B} \models Q^{\mathbf{A}}$, i.e., the structure $\mathbf{B}$ satisfies the canonical query $Q^{\mathbf{A}}$ of $\mathbf{A}$.*

To illustrate this result, recall that an undirected graph $\mathbf{G} = (V, E)$ is 3-colorable if and only if there is a homomorphism from $\mathbf{G}$ to $\mathbf{K}_3$. Consequently, Theorem 1 implies that $\mathbf{G}$ is 3-colorable if and only if $\mathbf{K}_3$ satisfies the canonical query $Q^{\mathbf{G}}$ of $\mathbf{G}$.

We say that two σ-structures $\mathbf{A}$ and $\mathbf{B}$ are *homomorphically equivalent* if there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$ and a homomorphism h' from $\mathbf{B}$ to $\mathbf{A}$. We write $\mathbf{A} \sim_h \mathbf{B}$ to denote that $\mathbf{A}$ is homomorphically equivalent to $\mathbf{B}$. Clearly, $\sim_h$ is an equivalence relation on the class of all finite σ-structures. Moreover, Theorem 1 implies that $\sim_h$ can be characterized in terms of logical equivalence.

Corollary 2. *The following are equivalent for finite σ-structures $\mathbf{A}$ and $\mathbf{B}$.*

1. *$\mathbf{A} \sim_h \mathbf{B}$.*
2. *$Q^{\mathbf{A}} \equiv Q^{\mathbf{B}}$, i.e., $Q^{\mathbf{A}}$ is is logically equivalent to $Q^{\mathbf{B}}$.*

2.2 Datalog, Pebble Games, and Constraint Satisfaction

Datalog is a database query language that can be succinctly described as logic programming without function symbols. More formally, a *Datalog program* is a finite set of rules of the form $t_0 \leftarrow t_1, \ldots, t_m$, where each t_i is an atomic formula $R(x_1, \ldots, x_n)$. The left-hand side of each rule is called the *head* of the rule, while the right-hand side is called the *body*. In effect, the body of each rule is a conjunctive query such that each variable occurring in the body, but not in the head, is existentially quantified. The relational predicates that occur in the heads of the rules are the *intensional database* predicates (IDBs), while all others are the *extensional database* predicates (EDBs). One of the IDBs is designated as the *goal* of the program. Note that IDBs may occur in the bodies of rules and, thus, a Datalog program is a recursive specification of the IDBs with semantics obtained via least fixed-points of monotone operators, see [Ull89]. Each Datalog program defines a query which, given a set of EDB predicates, returns the value of the goal predicate. If the goal predicate is 0-ary, then the program defines a Boolean query. Note that a Datalog query is computable in polynomial time, since the bottom-up evaluation of the least fixed-point of the program terminates within a polynomial number of steps (in the size of the given EDBs), see [Ull89]. Thus, expressibility in

Datalog is a sufficient condition for tractability of a query. As an example, NON-2-COLORABILITY is definable by the goal predicate Q of the Datalog program below, which asserts that the graph $\mathbf{G} = (V, E)$ contains a cycle of odd length:

$$\begin{aligned} P(x,y) &:- \ E(x,y) \\ P(x,y) &:- \ P(x,z), E(z,w), E(w,y) \\ Q &:- \ P(x,x) \end{aligned}$$

A key parameter in analyzing Datalog programs is the number of variables used. For every positive integer k, let k-Datalog be the collection of all Datalog programs in which the body of every rule has at most k distinct variables and also the head of every rule has at most k variables (the variables of the body may be different from the variables of the head). For instance, the preceding example shows that NON-2-COLORABILITY is definable by a 4-Datalog program (in fact, it is also definable by a 3-Datalog program).

If $\mathcal{A}$ is a class of σ-structures and $\mathbf{B}$ is a σ-structure, then $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is the complement (relative to $\mathcal{A}$) of $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$, i.e., it is the class of all σ-structures $\mathbf{A} \in \mathcal{A}$ such that there is *no* homomorphism h from $\mathbf{A}$ to $\mathbf{B}$. Feder and Vardi [FV98] showed that the tractability of many constraint satisfaction problems of the form $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is due to the fact that $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is expressible in k-Datalog, for some positive integer k. In other words, in many cases in which $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is tractable there is a positive integer k and a k-Datalog program P with a 0-ary goal predicate such that, for every σ-structure $\mathbf{A} \in \mathcal{A}$, we have that P on $\mathbf{A}$ evaluates to "true" iff $\mathbf{A} \notin \mathrm{CSP}(\mathcal{A}, \mathbf{B})$. A concrete instance of this phenomenon is 2-COLORABILITY, since it is the same decision problem as $\mathrm{CSP}(\mathcal{G}, \mathbf{K_2})$, where $\mathcal{G}$ is the class of all undirected graphs and $\mathbf{K}_2$ is the undirected graph consisting of a single edge. It should be pointed out that, when linking tractability of constraint satisfaction problems with definability in Datalog, it is necessary to consider the complement $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ of $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$, because $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ itself cannot be definable in Datalog. The reason is that Datalog-definable queries are *monotone*, in the sense that they are preserved under the addition of tuples in the relations of the input, while $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ lacks this monotonicity property.

It is well known that the expressive power of some of the main logical formalisms, including first-order logic and second-order logic, can be analyzed using certain combinatorial two-person games. In particular, the expressive power of k-Datalog can be analyzed using *existential k-pebble games*, which were introduced by Kolaitis and Vardi [KV95] in the context of database theory. These games are played between two players, the *Spoiler* and the *Duplicator*, on two σ-structures $\mathbf{A}$ and $\mathbf{B}$ according to the following rules: a round of the game consists of k moves of each player; on the i-th move of a round, $1 \leq i \leq k$, the Spoiler places a pebble on an element a_i of A, and the Duplicator responds by placing a pebble on an element b_i of B. At the end of the round, if the mapping $a_i \mapsto b_i$, $1 \leq i \leq k$, is not a homomorphim between the corresponding substructures of $\mathbf{A}$ and $\mathbf{B}$ induced by $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_k\}$, then the Spoiler wins the game. Otherwise, the Spoiler removes one or more pebbles, and a new round of the game begins. The Duplicator wins the existential k-pebble game if he has a *winning strategy*, i.e., a systematic way that allows him to sustain playing "forever", so that the Spoiler can never win a round of the game.

In [KV00a], it was shown that existential k-pebble games can be used to characterize when $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is expressible in k-Datalog. Moreover, in [KV00b] it was pointed out that there is a tight connection between existential k-pebble games and strong k-consistency properties of constraint satisfaction problems. Recall that a CSP-instance is *strongly k-consistent* if, for every $i < k$, every partial solution on $i-1$ variables can be extended to a partial solution on i variables. Moreover, the statement "strong k-consistency can be established for a CSP-instance" means that additional constraints can be added, so that the resulting CSP-instance is strongly k-consistent and has the same solutions as the original one (see [Dec92] for the precise definitions). The following is the key link between existential pebble games and strong consistency properties: given two σ-structures $\mathbf{A}$ and $\mathbf{B}$, it is possible to establish strong k-consistency for the CSP-instance associated with $\mathbf{A}$ and $\mathbf{B}$ ("is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$?") if and only if the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. By combining the results in [KV00a,KV00b], we obtain several different characterizations of when $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is expressible in k-Datalog. Before stating these characterizations we note that, whenever we write "strong k-consistency can be established for two σ-structures $\mathbf{A}$ and $\mathbf{B}$", we mean that strong k-consistency can be established for the CSP-instance associated with $\mathbf{A}$ and $\mathbf{B}$ (see [KV00b] for the formal definition of establishing strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$).

Theorem 3. [KV00a,KV00b] *Assume that $\mathcal{A}$ is a class of σ-structures, $\mathbf{B}$ is a σ-structure, and k is a positive integer. Then the following statements are equivalent.*

1. $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ *is expressible in k-Datalog.*
2. $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ *consists precisely of all structures $\mathbf{A} \in \mathcal{A}$ such that the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$.*
3. *For every structure $\mathbf{A} \in \mathcal{A}$, if the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.*
4. *For every structure $\mathbf{A} \in \mathcal{A}$, if strong k-consistency can be established for $\mathbf{A}$ and $\mathbf{B}$, then there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.*

When applied to our running example of 2-COLORABILITY, the preceding Theorem 3 implies (among other things) that a graph $\mathbf{G}$ is 2-colorable if and only the Duplicator wins the existential 4-pebble game on $\mathbf{G}$ and $\mathbf{K}_2$. As we will see in the sequel, Theorem 3 is a useful tool for determining when a tractable case of the constraint satisfaction problem is actually definable in Datalog. In addition, it reveals that close links exists between concepts in artificial intelligence and concepts in database theory and logic.

3 Bounded Treewidth and Datalog

Through the efforts of several different researchers, it has been established that many NP-complete problems on graphs become tractable when the input graphs are assumed to have a "tree-like" structure (see [DF99]). The property of being "tree-like" is formalized using the concept of the *treewidth* of a graph [DF99] or, more generally, the concept of the *treewidth* of a structrure, which is defined as follows [FV98]. A *tree decomposition* of a σ-structure $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$ is a labeled tree T such that:

1. Every node of T is labeled by a non-empty subset of V.
2. For every relation $R_i^{\mathbf{A}}$ and every tuple $(a_1, \ldots, a_n) \in R_i^{\mathbf{A}}$, there is a node of T whose label contains $\{a_1, \ldots, a_n\}$.
3. For every $a \in A$, the set X of nodes of T whose labels include a is a subtree of T.

The *width* of a tree decomposition T is the maximum cardinality of a label in T minus 1. The *treewidth* of a σ-structure $\mathbf{A} = (A, R_1^{\mathbf{A}}, \ldots, R_m^{\mathbf{A}})$ is the smallest positive integer k such that $\mathbf{A}$ has a tree decomposition of width k. Several algorithmic problems, including 3-COLORABILITY, that are NP-complete when arbitrary structures are allowed as inputs become solvable in polynomial time, if the inputs are restricted to be structures of treewidth less than k. We write $\mathcal{T}^k(\sigma)$ to denote the class of all finite σ-structures of treewidth less than k. We also write $\mathcal{T}^k$ to denote the class of all structures of treewidth less than k, i.e., $\mathcal{T}^k$ is the union of the classes $\mathcal{T}^k(\sigma)$ over all vocabularies σ.

Dechter and Pearl [DP89] and Freuder [Fre90] were the first to show that bounded treewidth is "an island of tractability" for constraint satisfaction problems. In terms of the notation used here, this means that $\mathrm{CSP}(\mathcal{T}^k, \mathcal{F})$ is solvable in polynomial time, where $\mathcal{F}$ is the class of all finite structures. In [KV00a] a different proof of the tractability of $\mathrm{CSP}(\mathcal{T}^k, \mathcal{F})$ was obtained by establishing a connection between bounded treewidth and definability in a certain fragment L^k of first-order logic with k variables. We now describe this fragment and the connection with bounded treewidth.

Let $x_1, \ldots, x_k$ be distinct first-order variables. We write L^k to denote the collection of first-order formulas over a vocabulary σ defined by the following conditions:

1. every atomic formula of σ with variables among $x_1, \ldots, x_k$ is an L^k-formula;
2. if φ and ψ are L^k-formulas, then $(\varphi \wedge \psi)$ is an L^k-formula;
3. if φ is an L^k-formula, then $(\exists x_i \varphi)$ is an L^k-formula, where $1 \leq i \leq k$.

Note that, although L^k has k distinct variables only, each variable may be reused again and again in an L^k-formula, so that there is no a priori bound on the number of occurrences of variables in L^k-formulas. Reusing variables is the key technique for showing that the expressive power of L^k is not as limited as it may initially appear. For exaple, by judiciously reusing variables, one can show that for every positive integer n, the property "there is a path of length n from a to b" is definable by an L^3-formula.

If $\mathbf{A}$ is a σ-structure with n elements in its universe, then clearly the canonical query $Q^{\mathbf{A}}$ is definable by an L^n-formula. In general, the number of variables needed to define $Q^{\mathbf{A}}$ cannot be reduced. Specifically, if $\mathbf{A}$ is the complete graph $\mathbf{K}_n$ with n nodes, then it can be shown that the canonical query $Q^{\mathbf{A}}$ is not definable by any L^m formula, for $m < n$. The state of affairs is different, however, if $\mathbf{A}$ has bounded treewidth.

Lemma 4. [KV00a] *If $\mathbf{A}$ is a σ-structure of treewidth less than k, then the canonical query $Q^{\mathbf{A}}$ is definable by an L^k-formula, which can be constructed in polynomial time.*

A proof that the constraint satisfaction problem $\mathrm{CSP}(\mathcal{T}^k, \mathcal{F})$ is solvable in polynomial time can be obtained by combining the above result with the fact that the evaluation problem for L^k-formulas is solvable in polynomial time [Var95]. Our goal now is to further explore the connection between bounded treewidth and definability in logics with a bounded number of variables. As described in Section 2, Feder and Vardi [FV98] showed that definability in Datalog provides a unifying explanation for the tractability of constraint satisfaction problems of the form $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$, where $\mathcal{A}$ is a class of σ-structures and $\mathbf{B}$ is a fixed σ-structure. The next result shows that definability in Datalog is also the reason for the tractability of constraint satisfaction problems on constraints of bounded treewidth.

Theorem 5. *Assume that k is a positive integer, σ is a vocabulary, and $\mathbf{B}$ is a σ-structure. Then $\neg\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathbf{B})$ is in k-Datalog, where $\mathcal{T}^k(\sigma)$ the class of all σ-structures of treewidth less than k.*

Proof: (Sketch) In view of Theorem 3, it suffices to show that if $\mathbf{A}$ and $\mathbf{B}$ are two σ-structures such that $\mathbf{A}$ is of treewidth $< k$ and the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$. Let $\mathbf{A}$ and $\mathbf{B}$ be two such structures and consider the canonical conjunctive query $Q^{\mathbf{A}}$ of $\mathbf{A}$. Since the treewidth of $\mathbf{A}$ is less than k, Lemma 4 implies that $Q^{\mathbf{A}}$ is definable by a sentence ψ of L^k. In particular, ψ is a sentence of $\exists\mathrm{FO}^k_+$, which is the fragment of first-order logic with k variables $x_1, \ldots, x_k$ that contains all atomic formulas involving these variables and is closed under conjunction, disjunction, and existential quantification over these variables. As shown in [KV95], there is a close connection between existential k-pebble games and preservation of $\exists\mathrm{FO}^k_+$-formulas. Specifically, if a σ-structure $\mathbf{A}$ satisfies an $\exists\mathrm{FO}^k_+$-sentence φ and if the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{B}$ satisfies the sentence φ as well. Consider the σ-structures $\mathbf{A}$ and $\mathbf{B}$ at hand. Clearly, $\mathbf{A} \models \psi$, because every structure satisfies its canonical query. Since the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, it follows that $\mathbf{B} \models \psi$, which means that $\mathbf{B} \models Q^{\mathbf{A}}$. By Theorem 1, a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ exists. ▮

Using Theorem 5, we now derive several additional results concerning the connections between constraint satisfaction, bounded treewidth, and definability in logics with a bounded number of variables. The first one follows from Theorems 3 and 5.

Corollary 6. *Assume that $k \geq 2$, $\mathbf{A}$ is a σ-structure of treewidth less than k, and $\mathbf{B}$ is an arbitrary σ-structure. Then the following statements are equivalent:*

1. *There is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.*
2. *The Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$.*
3. *Strong k-consistency can be established on $\mathbf{A}$ and $\mathbf{B}$.*

Consequently, determining whether strong k-consistency can be established is a sound and complete polynomial-time algorithm for $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$.

The typical use of strong k-consistency properties in constraint satisfaction problems is to try to establish strong k-consistency for a k that is sufficiently large to guarantee *global consistency*, which is the property that every partial solution can be extended to a solution (see [Dec92]). Corollary 6 yields a different use of strong k-consistency as a sound and complete algorithm for constraint satisfaction problems, when the constraints are of treewidth less than k. Although this result seems to be implicit in other published work, we have not been able to locate an explicit reference to it.

In general, expressibility in k-Datalog is a sufficient condition for tractability of $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$, but it does not provide a method for finding a solution to an instance of

$\mathrm{CSP}(\mathcal{A}, \mathbf{B})$, if one exists. This difficulty, however, can be overcome if more stringent definability conditions are satisfied. Specifically, [KV00b] introduced the concept of k-*locality* and showed that it is a sufficient condition for the backtrack-free construction of solutions to constraint satisfaction problems, if such solutions exist.

Let k be a positive integer, $\mathcal{A}$ a class of σ-structures, and $\mathbf{B}$ a σ-structure. We say that $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is k-*local* if $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B}^*)$ is in k-Datalog for every expansion $\mathbf{B}^*$ of $\mathbf{B}$ with constants, that is, for every expansion of $\mathbf{B}$ obtained by augmenting $\mathbf{B}$ with a finite sequence of distinguished elements from its universe. Such an expansion can be also viewed as a structure over a relational vocabulary σ^* in which unary relational symbols are used to encode the distinguished elements that form the expansion. We say that $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$ is k-*local* if $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is k-local, for every structure $\mathbf{B} \in \mathcal{B}$.

Theorem 7. [KV00b] *If $\mathrm{CSP}(\mathcal{A}, \mathcal{B})$ is k-local, then there is polynomial-time backtrack-free algorithm such that, given $\mathbf{A} \in \mathcal{A}$ and $\mathbf{B} \in \mathcal{B}$, it finds a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, if one exists, or determines that no such homomorphism exists, otherwise.*

This backtrack-free algorithm builds a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ in a sequence of steps; in each step, one tests whether strong k-consistency can be established for progressively longer expansions $\mathbf{A}^*$ and $\mathbf{B}^*$ of $\mathbf{A}$ and $\mathbf{B}$. Since $\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is k-local, if strong k-consistency can be established for some such expansions, then a homomorphism between these expansions is guaranteed to exist, which means that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ mapping the distinguished elements of $\mathbf{A}$ to the corresponding distinguished elements of $\mathbf{B}$. Consequently, the algorithm can proceed and construct longer expansions of $\mathbf{A}$ and $\mathbf{B}$ without backtracking, until every element of $\mathbf{A}$ is a distinguished element. Notice that this algorithm makes a quadratic number of calls to the test of whether strong k-consistency can be established.

Corollary 8. *$\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ is k-local, for every $k \geq 2$ and every σ.*

Proof: (Sketch) If a σ-structure $\mathbf{A}$ has treewidth less than k, then every expansion of it with constants also has treewidth less than k. To see this notice that each such expansion amounts to augmenting the vocabulary with unary predicates, and the addition of unary predicates does not change the treewidth of a structure. The result now follows immediately from Theorem 5. ∎

A different polynomial-time backtrack-free algorithm for $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ is known in the literature. Specifically, Section 1.4.2 of Hooker [Hoo97]) contains a description of a "zero-step lookahead" algorithm for constructing a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, where $\mathbf{A}$ is of treewidth $< k$. This algorithm is based on Freuder's [Fre90] result that the treewidth of a graph coincides with its *induced width*. Unlike the backtrack-free algorithm based on k-locality, the zero-step lookahead algorithm entails just a single initial test of whether strong k-consistency can be established. It requires, however, the efficient construction of an order of the universe of $\mathbf{A}$ (i.e., of the variables of the CSP-instance) of width $< k$. In turn, for each fixed k, such an order of the universe can be obtained in polynomial time from a tree decomposition of $\mathbf{A}$ of width $< k$, which has to be constructed first in polynomial time [Bod93].

So far, we have established that definability in k-Datalog provides an explanation for the tractability of $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathbf{B})$, where $\mathbf{B}$ is an arbitrary, but fixed, σ-structure. This, however, does not provide an explanation for the tractability of $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ in terms of definability in some tractable logical formalism with a bounded number of variables. Actually, there is a good reason for this, because the monotonicity properties of Datalog, imlpy that neither $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ nor $\neg\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ are expressible in Datalog. There is, however, a well-known logical formalism that is more powerful than Datalog and provides an explanation for the tractability of $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$. Specifically, *least fixed-point logic* LFP is the extension of first-order logic with least fixed-points of positive first-order formulas. Datalog can be viewed as a fragment of LFP, since Datalog queries are definable using least fixed-points of positive existential first-order formulas. Least fixed-point logic has found numerous applications to database theory and descriptive complexity theory, because of its close connections to polynomial-time computability ([Var82,Imm86,Imm99]). In particular, every LFP-definable query is also computable in polynomial-time. The next result shows that the tractability of $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$. can be explained via definability in LFP with a bounded number of variables.

Corollary 9. $\mathrm{CSP}(\mathcal{T}^k(\sigma), \mathcal{F}(\sigma))$ *is expressible in* LFP^{2k}*, for every* $k \geq 2$ *and every* σ*, where* LFP^{2k} *is the collection of all* LFP*-formulas with at most* $2k$ *distinct variables.*

Proof: (Hint) The result is derived by combining Corollary 6 with the fact that determining the winner in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$ is expressible in LFP^{2k}, when both structures $\mathbf{A}$ and $\mathbf{B}$ are part of the input (see [KV00b]). ∎

4 Bounded Treewidth and Homomorphic Equivalence

If σ is a vocabulary and $\mathcal{A}$ is a class of σ-structures, then we write $\mathcal{H}(\mathcal{A})$ to denote the class of all σ-structures that are homomorphically equivalent to some structure in $\mathcal{A}$. The first result of this section asserts intuitively that definability of constraint satisfaction problems in Datalog can be extended from a class $\mathcal{A}$ to the class $\mathcal{H}(\mathcal{A})$.

Proposition 10. *Let* $\mathcal{A}$ *be a class of* σ*-structures,* $\mathbf{B}$ *a* σ*-structure, and* k *a positive integer. If* $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ *is expressible in* k*-Datalog, then also* $\neg\mathrm{CSP}(\mathcal{H}(\mathcal{A}), \mathbf{B})$ *is expressible in* k*-Datalog.*

Proof: (Sketch) In view of Theorem 3, it suffices to show that if $\mathbf{A}$ is a structure in $\mathcal{H}(\mathcal{A})$ such that the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then there is a homomorphis h from $\mathbf{A}$ to $\mathbf{B}$. Assume that $\mathbf{A}$ is such a structure and let $\mathbf{A}'$ be a structure in $\mathcal{A}$ that is homomorphically equivalent to $\mathbf{A}$. This means that there is a homomorphism h_1 from $\mathbf{A}$ to $\mathbf{A}'$, and a homomorphism h_2 from $\mathbf{A}'$ to $\mathbf{A}$. By composing h_2 with the winning strategy for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, we obtain a winning strategy for the Duplicator in the existential k-pebble game on $\mathbf{A}'$ and $\mathbf{B}$. Since $\mathbf{A}'$ is in $\mathcal{A}$ and $\neg\mathrm{CSP}(\mathcal{A}, \mathbf{B})$ is expressible in k-Datalog, Theorem 3 implies that there is a homomorphism h' from $\mathbf{A}'$ to $\mathbf{B}$. Consequently, the composition $h = h_1 \circ h'$ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. ∎

By combining Theorem 5 with Proposition 10, we obtain the following result.

Corollary 11. *Assume that $k \geq 2$ and σ is a vocabulary.*

1. *If $\mathbf{B}$ is a σ-structure, then $\neg\mathrm{CSP}(\mathcal{H}(\mathcal{T}^k(\sigma)), \mathbf{B})$ is expressible in k-Datalog and, hence, it is in PTIME*
2. *$\mathrm{CSP}(\mathcal{H}(\mathcal{T}^k(\sigma)), \mathcal{F}(\sigma))$ is expressible in LFP^{2k} and, hence, it is in PTIME.*

Corollary 11 shows that the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$, give rise to larger "islands of tractability" for constraint satisfaction that those obtained from the classes $\mathcal{T}^k(\sigma)$ of structures of treewidth less than k. In what follows, we will show that the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$, possess also algebraic and logical characterizations that tie together some of the key concepts studied here. To establish this result we need to first bring the concept of a *core* of a structure into the picture.

Let $\mathbf{A}$ be a σ-structure. A substructure $\mathbf{B}$ of $\mathbf{A}$ is called a *core* of $\mathbf{A}$ if there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$, but, for every proper substructure $\mathbf{B}'$ of $\mathbf{B}$, there is no homomorphism from $\mathbf{A}$ to $\mathbf{B}'$. A σ-structure $\mathbf{A}$ is a *core* if it is its own core. Altough the study of cores originated in graph theory, the concept has found applications to database theory, as cores play an important role in conjunctive-query processing and optimization (see [CM77]). The following are some well known and easy to establish facts about cores (see [HN92]): (1) Every finite σ-structure $\mathbf{A}$ has a core; (2) If $\mathbf{B}$ is a core of $\mathbf{A}$, then $\mathbf{A}$ is homomorphically equivalent to $\mathbf{B}$; (3) If both $\mathbf{B}$ and $\mathbf{B}'$ are cores of $\mathbf{A}$, then $\mathbf{B}$ is isomorphic to $\mathbf{B}'$. In view of the last fact, we write $\mathrm{core}(\mathbf{A})$ for the unique (up to isomorphism) core of $\mathbf{A}$.

Let us consider some examples that illustrate these concepts and facts. First, the complete undirected graph $\mathbf{K}_2$ with two elements (i.e., the graph that consists of a single edge) is a core. Moreover, an undirected non-empty graph $\mathbf{G}$ has $\mathbf{K}_2$ as its core if and only if $\mathbf{G}$ is 2-colorable. Note that, for $k \geq 3$, this equivalence does not immediately extend to k-colorable graphs and to $\mathbf{K}_k$, because a k-colorable graph need not contain the complete undirected graph $\mathbf{K}_k$ as a subgraph. It is easy to see, however, that for every $k \geq 3$, an undirected graph $\mathbf{G}$ is k-colorable if and only if $\mathbf{K}_k$ is the core of $\mathbf{G} \oplus \mathbf{K}_k$, where $\mathbf{G} \oplus \mathbf{K}_k$ is the *direct sum* of $\mathbf{G}$ and $\mathbf{K}_k$. We are now ready to state and prove the promised characterizations of $\mathcal{H}(\mathcal{T}^k(\sigma))$.

Theorem 12. *Let k be a positive integer, σ a vocabulary, and $\mathbf{A}$ a σ-structure. Then the following statements are equivalent.*

1. *$\mathbf{A} \in \mathcal{H}(\mathcal{T}^k(\sigma))$.*
2. *$\mathrm{core}(\mathbf{A})$ has treewidth less than k.*
3. *The canonical conjunctive query $Q^{\mathbf{A}}$ is logically equivalent to an L^k-formula.*

Proof: (Sketch) We proceed in a round robin fashion. Assume that $\mathbf{A}$ is homomorphically equivalent to a σ-structure $\mathbf{A}'$ of treewidth $< k$. Since $\mathbf{A}$ and $\mathbf{A}'$ are homomorphically equivalent, it is easy to see that $\mathrm{core}(\mathbf{A})$ is isomorphic to $\mathrm{core}(\mathbf{A}')$. At the same time, $\mathrm{core}(\mathbf{A}')$ has treewidth less than k, since it is a substructure of a structure having treewidth less than k. If $\mathrm{core}(\mathbf{A})$ has treewidth less than k, then Lemma 4 implies that the canonical query $Q^{\mathrm{core}(\mathbf{A})}$ is logically equivalent to an L^k-formula. Since $\mathbf{A}$ is homomorphically equivalent to $\mathrm{core}(\mathbf{A})$, Corollary 2 implies that $Q^{\mathbf{A}}$ is logically equivalent to $Q^{\mathrm{core}(\mathbf{A})}$. Finally, assume that the canonical query $Q^{\mathbf{A}}$ is logically equivalent to an L^k-sentence ψ. As pointed out in [KV00b, Remark 5.3], if ψ is a an L^k-sentence, then one can construct a σ-structure $\mathbf{B}$ of treewidth less than k such that the canonical query $Q^{\mathbf{B}}$ is logically equivalent to ψ. Consequently, Corollary 2 implies that $\mathbf{A}$ is homomorphically equivalent to such a structure $\mathbf{B}$. ∎

It is well known that, for each positive integer k, there is a polynomial-time algorithm for determining whether a given structure has treewidth less than k [Bod93]. In other words, for each fixed k, membership in the class $\mathcal{T}^k(\sigma)$ can be tested in polynomial time. Our next result shows that, unfortunately, the state of affairs is dramatically different for the classes $\mathcal{H}(\mathcal{T}^k(\sigma))$, $k \geq 2$.

Theorem 13. *For every $k \geq 2$ and every vocabulary σ containing at least one binary relation symbol, determining membership in $\mathcal{H}(\mathcal{T}^k(\sigma))$ is an NP-complete problem.*

Proof: (Sketch) We first show that if $k \geq 3$ and $\mathbf{G}$ is an undirected graph, then the following are equivalent:

1. $\mathbf{G}$ is k-colorable.
2. $\mathrm{core}(\mathbf{G} \oplus \mathbf{K}_k)$ has treewidth less than k.
3. $\mathbf{G} \oplus \mathbf{K}_k \in \mathcal{H}(\mathcal{T}^k(\sigma))$.

Indeed, if $\mathbf{G}$ is k-colorable, then $\mathrm{core}(\mathbf{G} \oplus \mathbf{K}_k) = \mathbf{K}_k$, which has treewidth $k-1$. If $\mathrm{core}(\mathbf{G} \oplus \mathbf{K}_k)$ has treewidth less than k, then $\mathbf{G} \oplus \mathbf{K}_k$ is certainly homomorphically equivalent to a graph of treewidth less than k, since every graph is homomorphically equivalent to its core. Finally, assume that $\mathbf{G}$ is homomorphically equivalent to a graph H having treewidth less than k. It is known that if a graph has treewidth less than k, then it is k-colorable (this is easy to see using the fact that a graph has treewidth less than k if and only if it is a partial k-tree - see [DF99]). Consequently, $\mathbf{G}$ is k-colorable, because it is homomorphically equivalent to a k-colorable graph.

Next, we consider the case $k = 2$. Let $\mathbf{T}$ be a directed tree. We will exhibit a polynomial-time reduction of $\mathrm{CSP}(\mathbf{T})$ to $\mathcal{H}(\mathcal{T}^2(\sigma))$. For every σ-structure $\mathbf{G}$, we define $\mathbf{G}^*$ to be $(\mathbf{G} \oplus \mathbf{T})$, if the Duplicator wins the existential 2-pebble game on $\mathbf{G}$ and $\mathbf{T}$, and $\mathbf{K_3}$, otherwise. Clearly, $\mathbf{G}^*$ can be constructed from $\mathbf{G}$ in polynomial time. We claim that, for every $\mathbf{G}$, $\mathbf{G} \in \mathrm{CSP}(\mathbf{T})$ if and only if $\mathbf{G}^* \in \mathcal{H}(\mathcal{T}^2(\sigma))$. Assume first that $\mathbf{G}$ is in $\mathrm{CSP}(\mathbf{T})$, which means that there is a homomorphism from $\mathbf{G}$ to $\mathbf{T}$. Consequently, the Duplicator wins the existential 2-pebble game on $\mathbf{G}$ and $\mathbf{T}$. It follows that $\mathbf{G}^* = (\mathbf{G} \oplus \mathbf{T})$ and that $\mathbf{G}^*$ is homomorphically equivalent to $\mathbf{T}$, which has treewidth less than 2. Conversely, assume that $\mathbf{G}^* \in \mathcal{H}(\mathcal{T}^2(\sigma))$. It follows that $\mathbf{G}^* = \mathbf{G} \oplus \mathbf{T}$ is homomorphically equivalent to a σ-structure $\mathbf{H}$ of treewidth less than 2, and that the Duplicator wins the existential 2-pebble game on $\mathbf{G}$ and $\mathbf{T}$. Therefore, the Duplicator also wins the existential 2-pebble game on $\mathbf{G} \oplus \mathbf{T}$ and $\mathbf{T}$. In turn and since $\mathbf{G} \oplus \mathbf{T}$ is homomorphically equivalent to $\mathbf{H}$, it follows that the Duplicator wins the existential 2-pebble game on $\mathbf{H}$ and $\mathbf{T}$. Since $\mathbf{H}$ has treewidth less than k, Corollary 6 implies that there is a homomorphism from $\mathbf{H}$ to $\mathbf{T}$; in turn, this implies that there is a homomorphism from $\mathbf{G} \oplus \mathbf{T}$ to $\mathbf{T}$. By restricting this homomorphism to $\mathbf{G}$, we obtain the desired homomorphism from $\mathbf{G}$ to $\mathbf{T}$. The NP-hardness follows from the existence of particular directed trees $\mathbf{T}$ such that $\mathrm{CSP}(\mathbf{T})$ is an NP-complete problem [GWW92,HNZ96]. ∎

Theorem 13 suggests that the logical characterization given in Theorem 12 for the class $\mathcal{H}(\mathcal{T}^k(\sigma))$ is not feasibly effective. It is natural therefore to ask whether the property "$\mathbf{A}$ has treewidth less than k" possesses a logical characterization that might also explain the complexity-theoretic difference between this property and the property "$\mathbf{A}$ is homomorphically equivalent to a structure of treewidth less than k". Clearly, any such characterization should involve a refinement of the property "the canonical query $Q^{\mathbf{A}}$ is logically equivalent to an L^k-formula". We now introduce such a refinement.

Let $\mathbf{A}$ be a σ-structure, let $Q^{\mathbf{A}}$ be the canonical query associated with $\mathbf{A}$. Here, we identify $Q^{\mathbf{A}}$ with its defining formula, i.e., we view $Q^{\mathbf{A}}$ as an existential first-order formula of the form $(\exists z_1)\cdots(\exists z_n)\varphi(z_1,\ldots,z_n)$, where $\varphi(z_1,\ldots,z_n)$ is a conjunction of atomic formulas over σ with variables among $z_1,\ldots,z_n$. We say that a first-order sentence ψ is a *rewriting* of $Q^{\mathbf{A}}$ if there is a finite sequence of formulas $\psi_1,\ldots,\psi_m$ such that ψ_1 is $Q^{\mathbf{A}}$, ψ_m is ψ, and each ψ_{i+1} is obtained from ψ_i by applying one of the following *rewrite rules*:

A-Rule: *Associativity* of conjunction is applied to subformulas of ψ.
C-Rule: *Commutatitivity* of conjunction is applied to subformulas of ψ.
$\exists$-**Rule:** A subformula of ψ_i of the form $(\exists x(\theta_1 \wedge \theta_2))$ is replaced by the formula $((\exists x\theta_1) \wedge \theta_2)$, provided the variable x is not free in θ_2.
R-Rule: A subformula of ψ of the form $(\exists x\theta)$ is replaced by the fomula $(\exists y)\theta[x/y])$, where y does not occur free in θ and $\theta[x/y]$ is obtained from θ by replacing all free occurrences of x in θ by y.

These four rewrite rules are routinely used in database query processing and optimization in order to transform queries to equivalent, but less-costly-to-evaluate, queries [Ull89]. The final result of this paper asserts that these rules can also be used to obtain a logical characterization of bounded treewidth.

Theorem 14. *Let k be a positive integer, σ a vocabulary, and $\mathbf{A}$ a σ-structure. Then the following statements are equivalent.*

1. *$\mathbf{A}$ has treewidth less than k.*
2. *There is an L^k-sentence ψ that is a rewriting of $Q^{\mathbf{A}}$.*

Moreover, if $\mathbf{A}$ has treewidth less than k, then such a rewriting can be constructed in time polynomial in the size of $\mathbf{A}$.

Proof: (Hint) If $\mathbf{A}$ has treewidth less than k, then one can construct in polynomial time a linear order of the universe A of $\mathbf{A}$ of induced width less than k, which means that every element of A has fewer than k smaller neighbors in the traingulation of the constraint graph of $\mathbf{A}$ (see [Fre90]). Using this linear order, it is possible to evaluate the canonical conjunctive query $Q^{\mathbf{A}}$ on every σ-structure $\mathbf{B}$ using intermediate conjunctive queries each of which has at most k variables. Specifically, this evaluation can be carried out by simulating the steps of the "bucket elimination algorithm" for constraint satisfaction in [Dec99]. In turn, each step of this simulation can be translated to rewriting steps that transform the canonical conjunctive query $Q^{\mathbf{A}}$ to an L^k-sentence. For the other direction, one can use the rewriting to build a tree decomposition of width less than k of a σ-structure isomorphic to $\mathbf{A}$. ∎

References

[Bod93] H.L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 226–234, 1993.

[CM77] A.K. Chandra and P.M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proc. 9th ACM Symp. on Theory of Computing*, pages 77–90, 1977.

[Dec92] R. Dechter. From local to global consistency. *Artificial Intelligence*, 55(1):87–107, May 1992.

[Dec99] R. Dechter. Bucket elimination: a unifying framework for reasoning. *Artificial Intelligence*, 113(1–2):41–85, 1999.

[DF99] R.G. Downey and M.R. Fellows. *Parametrized Complexity*. Springer-Verlag, 1999.

[DP89] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, pages 353–366, 1989.

[Fre90] E.C Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proc. AAAI-90*, pages 4–9, 1990.

[FV98] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM J. on Computing*, 28:57–104, 1998.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., 1979.

[GWW92] W. Gutjahr, E. Welzl, and G. Woeginger. Polynomial Graph Colorings. *Discrete Appl. Math.*, 35:29–46, 1992.

[HN90] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.

[HN92] P. Hell and J. Nešetřil. The core of a graph. *Discrete Math.*, 109:117–126, 1992.

[HNZ96] P. Hell, J. Nesetril, and X. Zhu. Complexity of Tree Homomorphism. *Discrete Appl. Math.*, 70:23–36, 1996.

[Hoo97] J.N. Hooker. Constraint satisfaction methods for generating valid cuts. In *Advances in computational and stochastic optimization, logic programming, and heuristic studies*, pages 1–30. Kluwer, 1997.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Imm99] N. Immerman. *Descriptive Complexity*. Springer, 1999.

[Jea98] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200(1–2):185–204, 1998.

[KV95] Ph. G. Kolaitis and M. Y. Vardi. On the expressive power of Datalog: tools and a case study. *Journal of Computer and System Sciences*, 51(1):110–134, August 1995.

[KV00a] Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, pages 302–332, 2000.

[KV00b] Ph.G. Kolaitis and M.Y. Vardi. A game-theoretic approach to constraint satisfaction. In *Proc. of the 17th Nat. Conf. on Artificial Intelligence (AAAI 2000)*, pages 175–181, 2000.

[Sch78] T.J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th ACM Symp. on Theory of Computing*, pages 216–226, 1978.

[Ull89] J. D. Ullman. *Database and Knowledge-Base Systems, Volumes I and II*. Computer Science Press, 1989.

[Var82] M. Y. Vardi. The complexity of relational query languages. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 137–146, 1982.

[Var95] M.Y. Vardi. On the complexity of bounded-variable queries. In *Proc. 14th ACM Symp. on Principles of Database Systems*, pages 266–76, 1995.