A Game-Theoretic Approach to Constraint Satisfaction

Phokion G. Kolaitis*
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064
kolaitis@cse.ucsc.edu
www.cse.ucsc.edu/~kolaitis

Moshe Y. Vardi†
Department of Computer Science
Rice University
Houston, TX 77005-1892
vardi@cs.rice.edu
www.cs.rice.edu/~vardi

Abstract

We shed light on the connections between different approaches to constraint satisfaction by showing that the main consistency concepts used to derive tractability results for constraint satisfaction are intimately related to certain combinatorial pebble games, called the existential k-pebble games, that were originally introduced in the context of Datalog. The crucial insight relating pebble games to constraint satisfaction is that the key concept of strong k-consistency is equivalent to a condition on winning strategies for the Duplicator player in the existential k-pebble game. We use this insight to show that strong k-consistency can be established if and only if the Duplicator wins the existential k-pebble game. Moreover, whenever strong k-consistency can be established, one method for doing this is to first compute the largest winning strategy for the Duplicator in the existential k-pebble game and then modify the original problem by augmenting it with the constraints expressed by the largest winning strategy. This basic result makes it possible to establish deeper connections between pebble games, consistency properties, and tractability of constraint satisfaction. In particular, we use existential k-pebble games to introduce the concept of k-locality and show that it constitutes a new tractable case of constraint satisfaction that properly extends the well known case in which establishing strong k-consistency implies global consistency.

Introduction and Summary of Results

Constraint satisfaction has occupied a prominent place in AI research since the 1970s. The importance of constraint satisfaction stems from the fact that a large number of fundamental algorithmic problems from different areas of artificial intelligence can be modeled as constraint-satisfaction problems (CSP) in a natural way. The input to 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. Since in general constraint satisfaction is NP-complete, a considerable amount of effort has been dedicated to the discovery of tractable cases of constraint satisfaction, see (Mackworth & Freuder 1993; Dechter 1992; Jeavons, Cohen, & Gyssens 1997). The aim of this line of investigation is to design efficient algorithms for special cases of constraint satisfaction and to develop useful heuristics for the general case.

One of the most fruitful approaches to coping with the intractability of constraint satisfaction has been the introduction and use of various *consistency* concepts that make explicit additional constraints implied by the original constraints. The connection between consistency properties and tractability was first described in (Freuder 1978; 1982). In a similar vein, (Dechter 1992; van Beek 1994; van Beek & Dechter 1997) investigated the relationship between *local consistency* and *global consistency*. Intuitively, local consistency means that any partial solution on a set of variables can be extended to a partial solution containing an additional variable, whereas global consistency means that any partial solution can be extended to a global solution. Note that if the inputs are such that local consistency implies global consistency, then there is a polynomial-time algorithm for constraint satisfaction; moreover, in this case a solution can be constructed via a backtrack-free search.

In recent years, researchers have also embarked on an ambitious project aiming to classify the currently known tractable cases of constraint satisfaction and ultimately identify all tractable cases of this problem. Specifically, in (Feder & Vardi 1999) two conditions are isolated and are shown to be sufficient for tractability of constraint satisfaction and to also provide a unifying framework for a large number of tractability results in the literature. The first of these conditions is expressibility in Datalog, the main query language for deductive database and knowledge-base systems, while the second condition is group-theoretic. A related unifying framework for tractability of constraint satisfaction has been developed by Jeavons et al. in a sequence of papers, including (Jeavons, Cohen, & Gyssens 1995; 1996; 1997); the key theme of this framework is that tractability is intimately connected to certain algebraic closure properties of the constraints. Although the above two frameworks are of distinctly different character, they turn out to have several points in common. In fact, certain tractable cases in the

*Work partially supported by NSF grants CCR-9610257 and CCR-9732041

†Work partially supported by NSF grant CCR-9700061

first framework turn out to coincide with certain tractable cases in the second framework. Furthermore, one of these cases also coincides with the case in which local consistency implies global consistency; thus, these three different approaches to constraint satisfaction meet at this point.

Our goal in this paper is to shed additional light on the connections between the different approaches to constraint satisfaction. As pointed out first by (Feder & Vardi 1999), constraint satisfaction can be identified with the *homomorphism problem* on relational structures: given two finite relational structures $\mathbf{A}$ and $\mathbf{B}$ over the same vocabulary, is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$?[1] Informally, the structure $\mathbf{A}$ represents the variables and the constrained tuples of variables, the structure $\mathbf{B}$ represents the values and the constraints, and the homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ are precisely the solutions to the instance of the constraint-satisfaction problem encoded by $\mathbf{A}$ and $\mathbf{B}$. Using this viewpoint, we show that the main consistency concepts mentioned above are intimately related to certain combinatorial pebble games on relational structures that were originally introduced in the context of Datalog. It is well known that the expressive power of several major logical formalisms, including first-order logic and second-order logic, can be analyzed using certain combinatortial two-person games, see (Ebbinghaus, Flum, & Thomas 1994). As regards Datalog, *existential k-pebble games* were introduced in (Kolaitis & Vardi 1995) and used to analyze the expressive power of Datalog. These games are played between two players, the *Spoiler* and the *Duplicator*, on two relational structures $\mathbf{A}$ and $\mathbf{B}$ according to the following rules: on the i-th move of a round of the game, $1 \leq i \leq k$, 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. The Spoiler wins the game at the end of that 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 $\mathbf{B}$. 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*, that is to say, a systematic way that allows him to sustain playing "forever", so that the Spoiler can never win a round of the game.

The crucial insight that relates pebble games to constraint satisfaction is that the key concept of *strong k-consistency* (Dechter 1992) is equivalent to a property of winning strategies for the Duplicator in the existential k-pebble game. Specifically, after giving the formal definition of a winning strategy, we point out that an instance of a constraint-satisfaction problem is strongly k-consistent if and only if the family of *all* partial homomorphims f with $|f| < k$ is a winning strategy for the Duplicator in the existential k-pebble game on the two relational structures that represent the given instance. The connection between pebble games and consistency properties, however, is deeper than just a mere reformulation of the concept of strong k-consistency. Indeed, as mentioned earlier, consistency properties underly the process of making explicit new constraints that are implied be the original constraints. A key technical step in this approach is the procedure known as "establishing strong k-consistency", which propagates the original constraints, adds implied constraints, and transforms a given instance of a constraint satisfaction problem to a strongly k-consistent instance with the same solution space (Cooper 1989; Dechter 1992). Here we show that strong k-consistency can be established if and only if the Duplicator wins the existential k-pebble game. Moreover, whenever strong k-consistency can be established, one method for doing this is to first compute the largest winning strategy for the Duplicator in the existential k-pebble game and then modify the original problem by augmenting it with the constraints expressed by the largest winning strategy; we also show that this method gives rise to the least constrained instance that establishes strong k-consistency and, in addition, satisfies a natural *coherence* property. By combining this result with earlier results in (Kolaitis & Vardi 1995; 1998) concerning the definability of the largest winning strategy, it follows that the algorithm for establishing strong k-consistency in this way (with k fixed) is actually expressible in least fixed-point logic; this strengthens the fact that strong k-consistency can be established in polynomial time, when k is fixed.

After this, we show that there are further connections between pebble games, consistency properties, and tractability of constraint satisfaction. If $\mathbf{B}$ is a fixed finite relational structure, then $\mathrm{CSP}(\mathbf{B})$ is the following non-uniform constraint-satisfaction problem: given a finite relational structure $\mathbf{A}$, is there a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$? Note that if $\mathbf{B}$ is the complete graph K_3 on three vertices, then $\mathrm{CSP}(\mathbf{B})$ is 3-COLORABILITY; thus, $\mathrm{CSP}(\mathbf{B})$ may very well be an NP-complete problem. It was shown in (Feder & Vardi 1999; Kolaitis & Vardi 1998) that existential k-pebble games can be used to characterize when $\mathrm{CSP}(\mathbf{B})$ is expressible in Datalog (from which it follows that $\mathrm{CSP}(\mathbf{B})$ is also solvable in polynomial time). Specifically, it was established that for every relational structure $\mathbf{B}$, the complement of $\mathrm{CSP}(\mathbf{B})$ is expressible by a Datalog program with k variables if and only if $\mathrm{CSP}(\mathbf{B})$ coincides with the collection of all relational structures $\mathbf{A}$ such that the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. Consequently, this is also equivalent to the following condition: $\mathrm{CSP}(\mathbf{B})$ coincides with the collection of all relational structures $\mathbf{A}$ such that establishing strong k-consistency on $\mathbf{A}$ and $\mathbf{B}$ implies that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Expressibility in Datalog is certainly a condition that gives rise to a large tractable case of non-uniform constraint satisfaction. It has the disadvantage, however, that it does not yield a method for finding a solution to an instance of $\mathrm{CSP}(\mathbf{B})$, if a solution exists. This should be contrasted with the special case of expressibility in Datalog in which $\mathrm{CSP}(\mathbf{B})$ has the property that establishing strong k-consistency implies global consistency. We call this property *global k-consistency*. In this case, given an intance of $\mathrm{CSP}(\mathbf{B})$, we can first detect the existence of a solution by establishing strong k-consistency and then we can easily construct a solution using a backtrack-free

[1]A *homomorphism* is a mapping from the domain of $\mathbf{A}$ to the domain of $\mathbf{B}$ such that every tuple in a relation of $\mathbf{A}$ is mapped to a tuple in the corresponding relation of $\mathbf{B}$.

search. Although this special case does not suffer from the above disadvantage of Datalog, its applicability is limited, since it turns out to be equivalent to a very stringent closure property of the relations of $\mathbf{B}$ (Feder & Vardi 1999; Jeavons, Cohen, & Cooper 1998). This state of affairs motivates the pursuit of tractable cases that interpolate between global k-consistency and expressibility in Datalog. To this effect, using k-pebble games, we introduce the concept of *k-locality* and show that it constitutes a new tractable case of non-uniform constraint satisfaction that is broader than global k-consistency, is expressible in Datalog, but does not suffer from the aforementioned disadvantage of expressibility in Datalog. In particular, we show that if $\mathrm{CSP}(\mathbf{B})$ is k-local, then a solution (if one exists) to a given instance of $\mathrm{CSP}(\mathbf{B})$ can be constructed in polynomial time via a backtrack-free search during which strong k-consistency is established for certain expansions of the given instance. Moreover, we show that if $\mathrm{CSP}(\mathbf{B})$ is k-local, then computing the largest winning strategy for the Duplicator in the existential k-pebble game is the *only* way to obtain an instance that establishes strong k-consistency and satisfies the coherence property mentioned earlier.

Consistency and Pebble Games

The standard terminology in AI formalizes an instance $\mathcal{P}$ of CSP as a triple $(V, D, \mathcal{C})$, consisting of a set V of variables, a set D of values, and a collection $\mathcal{C}$ of *constraints* $C_1, \ldots, C_q$, where each C_i is a pair (t, R) with t a tuple over V (i.e., a tuple of not necessarily distinct variables in V) and R is a relation on D of the same arity as $|t|$. Note that, without loss of generality, we may assume that all constraints (t, R_i) involving a tuple t have been consolidated to a single constraint (t, R). Thus, we can assume that each tuple t of variables occurs at most once in the collection $\mathcal{C}$. It is clear that every such instance $\mathcal{P}$ can be viewed as an instance of the homomorphism problem between two structures $\mathbf{A}_{\mathcal{P}}$ and $\mathbf{B}_{\mathcal{P}}$, where the universe of $\mathbf{A}_{\mathcal{P}}$ is V, the universe of $\mathbf{B}_{\mathcal{P}}$ is D, the relations of $\mathbf{B}_{\mathcal{P}}$ are the distinct relations R occurring in $\mathcal{C}$, and the relations of $\mathbf{A}_{\mathcal{P}}$ are defined as follows: for each relation R on D occurring in $\mathcal{C}$, we have the relation $R^{\mathbf{A}} = \{t : (t, R) \text{ is a constraint}\}$. We call $(\mathbf{A}_{\mathcal{P}}, \mathbf{B}_{\mathcal{P}})$ the *homomorphism instance* of $\mathcal{P}$. It is also clear that every instance of the homomorphism problem between two structures $\mathbf{A}$ and $\mathbf{B}$ can be viewed as a CSP instance $\mathrm{CSP}(\mathbf{A}, \mathbf{B})$ by simply "breaking up" each relation $R^{\mathbf{A}}$ on $\mathbf{A}$ as follows: we generate a constraint $(t, R^{\mathbf{B}})$ for each $t \in R^{\mathbf{A}}$. (and then consolidate constraints involving the same tuple of variables). We call $\mathrm{CSP}(\mathbf{A}, \mathbf{B})$ the *CSP instance* of $(\mathbf{A}, \mathbf{B})$ We will use both formalisms in this paper, as each has its own advantages.

The next definition contains the main concepts concerning existential k-pebble games.

Definition 1: Let k be a positive integer and let $\mathbf{A}$ and $\mathbf{B}$ be two relational structures over the same vocabulary with universes A and B respectively.

- A *k-partial homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a homomorphism from a substructure of $\mathbf{A}$ with at most k elements in its universe to a substructure of $\mathbf{B}$.
- A *winning strategy for the Duplicator in the existential k-pebble game on* $\mathbf{A}$ *and* $\mathbf{B}$ is a nonempty family $\mathcal{F}$ of k-partial homomorphisms having the following two properties:
  1. $\mathcal{F}$ is *closed under subfunctions*, which means that if $g \in \mathcal{F}$ and $f \subseteq g$, then $f \in \mathcal{F}$.
  2. $\mathcal{F}$ has the *k-forth property*, which means that for every $f \in \mathcal{F}$ with $|f| < k$ and every $a \in A$ on which f is undefined, there is a $g \in \mathcal{F}$ that extends f and is defined on a.
- A *configuration for the existential k-pebble game on* $\mathbf{A}$ *and* $\mathbf{B}$ is a $2k$-tuple $\overline{a}, \overline{b}$, where $\overline{a}$ and $\overline{b}$ are elements of A^k and B^k respectively such that if $a_i = a_j$, then $b_i = b_j$ (i.e., the correspondence $a_i \mapsto b_i$, $1 \le i \le k$, is a partial function from A to B, which we denote by $h_{\overline{a},\overline{b}}$).
- A *winning configuration for the Duplicator in the existential k-pebble game on* $\mathbf{A}$ *and* $\mathbf{B}$ is a configuration $\overline{a}, \overline{b}$ for this game such that $h_{\overline{a},\overline{b}}$ is a member of some winning strategy for the Duplicator in this game. We denote by $\mathcal{W}^k(\mathbf{A}, \mathbf{B})$ the set of all such configurations.

∎

The following facts turn out to be quite useful.

Proposition 2: *If $\mathcal{F}$ and $\mathcal{F}'$ are two winning strategies for the Duplicator in the existential k-pebble game on two structures $\mathbf{A}$ and $\mathbf{B}$, then also the union $\mathcal{F} \cup \mathcal{F}'$ is a winning strategy for the Duplicator. Hence, there is a largest winning strategy for the Duplicator in the existential k-pebble game, namely the union of all winning strategies, which is precisely*

$$\mathcal{H}^k(\mathbf{A}, \mathbf{B}) = \{h_{\overline{a},\overline{b}} : (\overline{a}, \overline{b}) \in \mathcal{W}^k(\mathbf{A}, \mathbf{B})\}.$$

Proof: The first part is obvious. For the second part, note that $\mathcal{H}^k(\mathbf{A}, \mathbf{B})$ is clearly a winning strategy for the Duplicator and contains every winning strategy as a subset, since every element h of a winning strategy gives rise to a winning configuration $\overline{a}, \overline{b}$ such that $h_{\overline{a},\overline{b}} = h$, where $\overline{a}$ is a list of all elements in the domain of h and $\overline{b}$ is a list of their images under h (the list may contain elements with repetitions, if the domain of h has fewer than k elements). ∎

The following lemma is a crucial definability result.

Lemma 3: (Kolaitis & Vardi 1998) *There is a positive first-order formula $\varphi(\overline{x}, \overline{y}, S)$, where $\overline{x}$ and $\overline{y}$ are k-tuples of variables, such that the complement of its least fixed-point on a pair $\mathbf{A}, \mathbf{B}$ of structures defines the set $\mathcal{W}^k(\mathbf{A}, \mathbf{B})$ of all winning configurations for the Duplicator in the existential k-pebble game on $\mathbf{A}, \mathbf{B}$.*

We now recall the concepts of *i-consistency* and *strong k-consistency*.

Definition 4: Let $\mathcal{P} = (V, D, \mathcal{C})$ be a CSP instance. $\mathcal{P}$ is *i-consistent* if for every $i-1$ variables $v_1, \ldots, v_{i-1}$, for every partial solution on these variables, and for every variable $v_i \notin \{v_1, \ldots, v_{i-1}\}$, there is a partial solution on the variables $v_1, \ldots, v_{i-1}, v_i$ extending the given partial solution on the variables $v_1, \ldots, v_{i-1}$. $\mathcal{P}$ is *strongly k-consistent* if it is i-consistent for every $i \le k$. ∎

A key insight is that strong k-consistency can be naturally recast in terms of existential k-pebble games.

Proposition 5: *Let $\mathcal{P}$ be a CSP instance, and let $(\mathbf{A}_{\mathcal{P}}, \mathbf{B}_{\mathcal{P}})$ be the associated homomorphism instance. $\mathcal{P}$ is strongly k-consistent if and only if the family of all k-partial homomorphisms from $\mathbf{A}_{\mathcal{P}}$ to $\mathbf{B}_{\mathcal{P}}$ is a winning strategy for the Duplicator in the existential k-pebble game on $\mathbf{A}_{\mathcal{P}}$ and $\mathbf{B}_{\mathcal{P}}$.*

Let us now recall the concept of *establishing strong k-consistency*, as defined, for instance, in (Cooper 1989; Dechter 1992). This concept has been defined rather informally in the literature to mean that, given an instance $\mathcal{P}$ of CSP, we associate an instance $\mathcal{P}'$ that has the following properties: (1) $\mathcal{P}'$ has the same set of variables and the same set of values as $\mathcal{P}$; (2) $\mathcal{P}'$ is strongly k-consistent; (3) $\mathcal{P}'$ is more constrained than $\mathcal{P}$; and (4) $\mathcal{P}$ and $\mathcal{P}'$ have the same space of solutions. The next definition formalizes the above concept in the context of the homomorphism problem.

Definition 6: Let $\mathbf{A}$ and $\mathbf{B}$ be two relational structures over a k-ary vocabulary σ (i.e., every relation symbol in σ has arity at most k). *Establishing strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$* means that we associate two relational structures $\mathbf{A}'$ and $\mathbf{B}'$ with the following properties:

1. $\mathbf{A}'$ and $\mathbf{B}'$ are structures over some k-ary vocabulary σ' (in general, different than σ); moreover, the universe of $\mathbf{A}'$ is the universe A of $\mathbf{A}$, and the universe of $\mathbf{B}'$ is the universe B of $\mathbf{B}$.
2. $\mathrm{CSP}(\mathbf{A}', \mathbf{B}')$ is strongly k-consistent.
3. if h is a k-partial homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$, then h is a k-partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$.
4. If h is a function from A to B, then h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if h is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$.

If the structures $\mathbf{A}'$ and $\mathbf{B}'$ have the above properties, then we say that *$\mathbf{A}'$ and $\mathbf{B}'$ establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$*. ∎

An instance $\mathcal{P}$ of CSP is *coherent* if every constraint (t, R) of $\mathcal{P}$ completely determines all constraints (u, Q) in which all variables occurring in u are among the variables of t. We formalize this concept as follows.

Definition 7: An instance $\mathbf{A}, \mathbf{B}$ of the homomorphism problem is *coherent* if its associated CSP instance $\mathrm{CSP}(\mathbf{A}, \mathbf{B})$ has the following property: for every constraint $(\overline{a}, R)$ of $\mathrm{CSP}(\mathbf{A}, \mathbf{B})$ and every tuple $\overline{b} \in R$, the mapping $h_{\overline{a},\overline{b}}$ is well defined and is a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$. ∎

Note that a CSP instance can be made coherent by polynomial-time constraint propagation.

The main result of this section is that strong k-consistency can be established precisely when the Duplicator wins the existential k-pebble game. Moreover, one method for establishing strong k-consistency is to first compute the largest winning strategy for the Duplicator in this game and then generate an instance of the constraint-satisfaction problem consisting of all the constraints embodied in the largest winning strategy. Furthermore, this method gives rise to the largest coherent instance that establishes strong k-consistency (and, hence, the least constrained such instance).

Theorem 8: *Let k be a positive integer, let σ be a k-ary vocabulary, and let $\mathbf{A}$ and $\mathbf{B}$ be two relational structures over σ with domains A and B, respectively. It is possible to establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$ if and only if $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$. Furthermore, if $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$, then the following sequence of steps gives rise to two structures $\mathbf{A}'$ and $\mathbf{B}'$ that establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$:*

1. *Compute the set $\mathcal{W}^k(\mathbf{A}, \mathbf{B})$.*
2. *Form the set $\mathcal{W}^k_*(\mathbf{A}, \mathbf{B})$ of all $2i$-tuples $(a_{j_1}, \ldots, a_{j_i}, b_{j_1}, \ldots, b_{j_i}) \in A^i \times B^i$, $1 \leq i \leq k$, that can be extended to a $2k$-tuple $(a_1, \ldots, a_k, b_1, \ldots, b_k) \in W^k(\mathbf{A}, \mathbf{B})$.*
3. *For every $i \leq k$ and for every i-tuple $\overline{a} \in A^i$, form the set $R_{\overline{a}} = \{\overline{b} \in B^i : (\overline{a}, \overline{b}) \in \mathcal{W}^k_*(\mathbf{A}, \mathbf{B})\}$.*
4. *Form the CSP instance $\mathcal{P}$ with A as the set of variables, B as the set of values, and $\{(\overline{a}, R_{\overline{a}}) : \overline{a} \in \bigcup_{i=1}^{k} A^i\}$ as the collection of constraints.*
5. *Let $(\mathbf{A}', \mathbf{B}')$ be the homomorphism instance of $\mathcal{P}$.*

In addition, the structures $\mathbf{A}'$ and $\mathbf{B}'$ obtained above constitute the largest coherent instance establishing strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$, i.e., if $(\mathbf{A}'', \mathbf{B}'')$ is another such coherent instance, then for every constraint $(\overline{a}, R)$ of $\mathrm{CSP}(\mathbf{A}'', \mathbf{B}'')$, we have that $R \subseteq R_{\overline{a}}$.

Proof: Suppose first that $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$. We now show that $\mathrm{CSP}(\mathbf{A}', \mathbf{B}')$ is strongly k-consistent. To see this, assume that g is a partial homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$ with domain $\{a_1, \ldots, a_i\}$, for some $i < k$, and c is an element of A. Let $b_j = g(a_j)$, $1 \leq j \leq i$, let $\overline{a} = (a_1, \ldots, a_i)$ and $\overline{b} = (b_1, \ldots, b_i)$. Since g is a partial homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$, it must be the case that $\overline{b} \in R_{\overline{a}}$, which in turn means that $\overline{a}, \overline{b}$ is a winning configuration for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. It follows that there is an element d of B such that $\overline{a}, c, \overline{b}, d$ is a winning configuration for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. In turn, this means that $\overline{b}, d \in R_{\overline{a},c}$. It is easy, however, to verify that $(\mathbf{A}', \mathbf{B}')$ is coherent and so the mapping $g \cup \{(c, d)\}$ is a partial homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$ extending g.

Next assume that h is a function from A to B. We have to show that h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if h is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$. Let $\overline{a} = (a_1, \ldots, a_k)$ be a k-tuple of elements from A and let $\overline{b} = (h(a_1), \ldots, h(a_k))$. Assume first that h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. In this case, we have that $\overline{a}, \overline{b}$ is a winning configuration for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, which in turn implies that $\overline{b} \in R_{\overline{a}}$, thus establishing that h is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$. In the other direction, if h is a homomorphism from $\mathbf{A}'$ to $\mathbf{B}'$, then $\overline{b} \in R_{\overline{a}}$, which means that $\overline{a}, \overline{b}$ is a winning configuration for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. In turn, this implies that if a relation of $\mathbf{A}$ is satisfied by a sequence of elements from $\overline{a}$, then the corresponding sequence of elements from $\overline{b}$ satisfies the corresponding relation on $\mathbf{B}$, thus establishing that h is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

Conversely, suppose that $\mathbf{A}'$ and $\mathbf{B}'$ establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$. Let $\mathcal{H}$ be the family of all k-partial homomorphisms from $\mathbf{A}'$ to $\mathbf{B}'$. By the definition of establishing strong k-consistency, $\mathcal{H}$ is also a family of k-partial homomorphisms from $\mathbf{A}$ to $\mathbf{B}$. Since, $\mathrm{CSP}(\mathbf{A}', \mathbf{B}')$ is strongly k-consistent, $\mathcal{H}$ has the k-forth property. But this means that the Duplicator has a winning strategy in the existential k-pebble game on $\mathbf{A}, \mathbf{B}$, which implies that $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$.

As mentioned earlier, $(\mathbf{A}', \mathbf{B}')$ is coherent. Assume that $(\mathbf{A}^*, \mathbf{B}^*)$ is another coherent instance establishing strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$. Let $(\overline{a}, R)$ be a constraint of $\mathrm{CSP}(\mathbf{A}^*, \mathbf{B}^*)$, and let $\overline{b} \in R$. Then the mapping $h_{\overline{a},\overline{b}}$ is a partial homomorphism from $\mathbf{A}^*$ to $\mathbf{B}^*$, which in turn implies that it is also a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$. It follows that $(\overline{a}, \overline{b}) \in \mathcal{W}^k_*(\mathbf{A}, \mathbf{B})$, and thus $\overline{b} \in R_{\overline{a}}$. ∎

The key step in the procedure described in Theorem 8 is the first step, in which the set $\mathcal{W}^k(\mathbf{A}, \mathbf{B})$ is computed. The other steps simply "re-format" $\mathcal{W}^k(\mathbf{A}, \mathbf{B})$. From Lemma 3, it follows that we can establish strong k-consistency by computing the least fixed-point of a positive first-order formula. This perspective should be contrasted with the efficient-implementation perspective in (Cooper 1989), the algebraic perspective described in (Güsgen & Ladkin 1995), and the chaotic-iteration perspective described in (Apt 1997).

One advantage of formalizing the concept of strong k-consistency in Definition 6 is that we can now address the computational complexity of establishing strong k-consistency. That is, how hard is it to determine whether it is possible to establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$, given two structures $\mathbf{A}$, $\mathbf{B}$ and a positive integer k? In view of Theorem 8, this key question is equivalent to asking how hard it is to test whether $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$. We conjecture that the exponential upper bound from (Kolaitis & Vardi 1995) is tight.

Conjecture: Checking whether $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$ for given structures $\mathbf{A}, \mathbf{B}$ and a positive integer k is EXPTIME-complete.

Note that a confirmation of this conjecture will explain why all known algorithms for establishing strong k-consistency are exponential in k (see (Cooper 1989; Dechter 1992)).

We can now relate the concept of strong k-consistency to the results in (Feder & Vardi 1999; Kolaitis & Vardi 1998) regarding Datalog and non-uniform CSP. *Datalog* is the language of database logic programming; it has received a tremendous amount of attention over the past two decades, see (Abiteboul, Hull, & Vianu 1995). 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 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 (Ullman 1989). 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 is a Boolean query, i.e., it either holds or does not. 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 (Ullman 1989). Thus, expressibility in Datalog is a sufficient condition for tractability of a query.

Let $\mathbf{B}$ be a relational structure over a vocabulary σ. Let $\neg\mathrm{CSP}(\mathbf{B})$ be the class of all structures $\mathbf{A}$ over the vocabulary σ such that there is no homomorphism h from $\mathbf{A}$ to $\mathbf{B}$. A unifying explanation for the tractability of many non-uniform $\mathrm{CSP}(\mathbf{B})$ problems is provided by showing that $\neg\mathrm{CSP}(\mathbf{B})$ is expressible in Datalog (Feder & Vardi 1999). That is, in many cases in which $\mathrm{CSP}(\mathbf{B})$ is tractable there is a Boolean Datalog program P such that for every structure $\mathbf{A}$ over σ, we have that $P(\mathbf{A})$ holds iff $\mathbf{A} \notin \mathrm{CSP}(\mathbf{B})$, A key parameter that shows up in this analysis is the number of variables used. For every positive integer n, 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).

Theorem 9: (Kolaitis & Vardi 1998) *Let* $\mathbf{B}$ *be a relational structure over a vocabulary* σ. $\neg\mathrm{CSP}(\mathbf{B})$ *is expressible in* k*-Datalog iff the following condition holds:*

> *For every structure* $\mathbf{A}$ *over* σ*, if the Duplicator wins the existential* k*-pebble game on* $\mathbf{A}$ *and* $\mathbf{B}$*, then there is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$.

We can now derive a relationship between k-Datalog and strong k-consistency.

Theorem 10: *Let* $\mathbf{B}$ *be a relational structure over a vocabulary* σ. $\neg\mathrm{CSP}(\mathbf{B})$ *is expressible in* k*-Datalog iff for every structure* $\mathbf{A}$ *over* σ*, establishing strong* k*-consistency for* $\mathbf{A}, \mathbf{B}$ *implies that there is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$.

Proof: Since the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$ if and only if $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$, the result follows from Theorems 8 and 9. ∎

Consistency and Locality

As mentioned in the introduction, expressibility in k-Datalog is a sufficient condition for tractability of $\mathrm{CSP}(\mathbf{B})$, but it does not provide a method for finding a solution to an instance of $\mathrm{CSP}(\mathbf{B})$, if one exists. In contrast, if $\mathrm{CSP}(\mathbf{B})$ has the global k-consistency property, (i.e., establishing stong k-consistency implies global consistency), then a solution to an instance of $\mathrm{CSP}(\mathbf{B})$ can be constructed via a backtrack-free search. Since the latter condition is of limited applicability, it is natural to pursue conditions that are of wider applicability and still yield a method for finding a solution efficiently, if one exists.

Definition 11: Let $\mathbf{B}$ be a structure over a relational vocabulary σ and let k be a positive integer. We say that $\mathrm{CSP}(\mathbf{B})$ is *k-local* if $\neg\mathrm{CSP}(\mathbf{B}^*)$ 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. Note that 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. ∎

The first result of this section yields a characterization of k-locality in terms of establishing strong k-consistency. Moreover, it asserts that k-locality has the property that there is a unique way to obtain a coherent instance establishing strong k-consistency.

Proposition 12: *Let $\mathbf{B}$ be a relational structure over a vocabulary σ. $\mathrm{CSP}(\mathbf{B})$ is k-local iff for every structure $\mathbf{A}$ over σ and every expansions $\mathbf{A}^*$ and $\mathbf{B}^*$ of $\mathbf{A}$ and $\mathbf{B}$ with constants, establishing strong k-consistency on $\mathbf{A}^*$ and $\mathbf{B}^*$ implies that there is a homomorphism from $\mathbf{A}^*$ to $\mathbf{B}^*$. Moreover, if $\mathrm{CSP}(\mathbf{B})$ is k-local, then the only way to obtain a coherent instance establishing strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$ is to compute the largest winning strategy for the Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$.*

Proof: The characterization of k-locality in terms of establishing strong k-consistency is an immediate consequence of Theorem 10. Assume that $(\mathbf{A}'', \mathbf{B}'')$ is a coherent pair of structures establishing strong k-consistency for $(\mathbf{A}, \mathbf{B})$. Let $(\overline{a}, R)$ be a constraint of $\mathrm{CSP}(\mathbf{A}'', \mathbf{B}'')$. From Theorem 8, it follows that $R \subseteq R_{\overline{a}}$, where $R_{\overline{a}}$ is the set of all tuples $\overline{b}$ such that $(\overline{a}, \overline{b}) \in \mathcal{W}^k_*(\mathbf{A}, \mathbf{B})$. For the other direction, if $\overline{b} \in R_{\overline{a}}$, then $(\overline{a}, \overline{b}) \in \mathcal{W}^k_*(\mathbf{A}, \mathbf{B})$ and so the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$ with pebbles placed on $\overline{a}$ and $\overline{b}$. Since $\mathrm{CSP}(\mathbf{B})$ is k-local, $\neg\mathrm{CSP}(\mathbf{B}, \overline{b})$ is expressible in Datalog. Consequently, by Theorem 9, it follows that there is a homomorphism h from $\mathbf{A}$ to $\mathbf{B}$ extending the partial homomorphism $a_i \mapsto b_i$, where a_i and b_i are the elements of A and B occuring in $\overline{a}$ and $\overline{b}$. Since $(\mathbf{A}'', \mathbf{B}'')$ establishes strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$, it follows that h is a homomorphism from $\mathbf{A}''$ to $\mathbf{B}''$. Thus, $\overline{b} \in R$, which establishes that $R = R_{\overline{a}}$. ∎

The next result presents the relationship between k-locality and the other tractable cases of non-uniform constraint satisfaction considered earlier. Moreover, it asserts that if $\mathrm{CSP}(\mathbf{B})$ is k-local, then there is a polynomial-time algorithm for finding a solution to a given instance of a $\mathrm{CSP}(\mathbf{B})$.

Theorem 13: *Let $\mathbf{B}$ be a relational structure over a vocabulary σ and let k be a positive integer.*

1. *If $\mathrm{CSP}(\mathbf{B})$ is k-local, then $\neg\mathrm{CSP}(\mathbf{B})$ is expressible in k-Datalog.*
2. *If $\mathrm{CSP}(\mathbf{B})$ has the global k-consistency property, then $\mathrm{CSP}(\mathbf{B})$ is k-local.*
3. *If $\mathrm{CSP}(\mathbf{B})$ is k-local, then there is a polynomial-time algorithm that, given a structure $\mathbf{A}$ over σ, finds a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, if one exists.*

Proof: (*Sketch*) The first two parts follow easily from the definitions, Theorem 8, and Proposition 12. For the third part, given a structure $\mathbf{A}$ over σ, one first checks whether $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$ to determine whether a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ exists. If $\mathcal{W}^k(\mathbf{A}, \mathbf{B}) \neq \emptyset$, then one can build a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ via a backtrack-free search that takes at most $O(n)$ steps; in each step, one has to test whether strong k-consistency can be established for progressively longer expansions $\mathbf{A}^*$ and $\mathbf{B}^*$ of $\mathbf{A}$ and $\mathbf{B}$ respectively. ∎

Note that if $\mathrm{CSP}(\mathbf{B})$ is k-local, then the algorithm for constructing a homomorphism is similar to the algorithm for constructing a homomorphism in the case where the global k-consistency property holds. The difference between these two algorithms is that in the latter case there is a single test in the beginning to determine whether it is possible to establish strong k-consistency for $\mathbf{A}$ and $\mathbf{B}$, whereas in the case of k-locality the test as to whether it is possible to establish strong k-consistency is repeatedly applied to the expansions $\mathbf{A}^*$ and $\mathbf{B}^*$ of $\mathbf{A}$ and $\mathbf{B}$ built during the backtrack-free search.

According to Theorem 13, the global k-consistency property implies k-locality. We can prove that this implication cannot be reversed. That is, k-locality is a tractable case of non-uniform constraint satisfaction that properly contains the case in which establishing strong k-consistency implies global consistency. For $k > 2$, let HORN k-SAT be the restriction of the satisfiability problem to k-CNF formulas in which every clause is Horn, i.e., it has at most one positive literal. It is easy to see that HORN k-SAT can be cast as a non-uniform CSP problem $\mathrm{CSP}(\mathbf{B}_k)$, where the universe of $\mathbf{B}_k$ is $\{0, 1\}$ and the relations of $\mathbf{B}_k$ encode the truth tables of Horn clauses with at most k literals. The proof of the next result will appear in the full paper.

Theorem 14: *Let $k > 2$ be a positive integer and let $\mathbf{B}_k$ be a structure that encodes HORN k-SAT. Then $\mathrm{CSP}(\mathbf{B}_k)$ is k-local, but there is no positive integer l such that $\mathrm{CSP}(\mathbf{B}_k)$ has the global l-consistency property.*

As mentioned in the introduction, the global k-consistency property is equivalent to a certain closure property of the relations of $\mathbf{B}$, (Feder & Vardi 1999; Jeavons, Cohen, & Cooper 1998). Since this closure property is decidable, it follows that there is an algorithm to decide whether, given a structure $\mathbf{B}$, the non-uniform constraint satisfaction problem $\mathrm{CSP}(\mathbf{B})$ has the global k-consistency property. In contrast, expressibility in k-Datalog in not known to be a decidable property. We also do not know whether k-locality is a decidable property. One way to attack this problem is to try to relate k-locality to a closure property, as in (Feder & Vardi 1999; Jeavons, Cohen, & Cooper 1998).

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of databases*. Addison-Wesley.

Apt, K. 1997. The essence of constraint propagation. *Theoretical Computer Science* 221(1–2):179–210.

Cooper, M. 1989. An optimal k-consistency algorithm. *Artificial Intelligence* 41(1):89–95.

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

Ebbinghaus, H.-D.; Flum, J.; and Thomas, W. 1994. *Mathematical Logic*. Springer-Verlag, 2nd edition.

Feder, T., and Vardi, M. 1999. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM J. on Computing* 28:57–104. Preliminary version in *Proc. 25th ACM Symp. on Theory of Computing*, May 1993, pp. 612–622.

Freuder, E. 1978. Synthesizing constraint expressions. *Communications of the ACM* 21(11):958–966.

Freuder, E. 1982. A sufficient condition for backtrack-free search. *Journal of the Association for Computing Machinery* 29(1):24–32.

Güsgen, H., and Ladkin, P. 1995. An algebraic approach to general Boolean constraints problem. Technical report, University of Bielfeld. RVS-RR-96-04.

Jeavons, P.; Cohen, D.; and Cooper, M. 1998. Constraints, consistency and closure. *Artificial Intelligence* 101(1-2):251–65.

Jeavons, P.; Cohen, D.; and Gyssens, M. 1995. A unifying framework for tractable constraints. In Montanari, U., and Rossi, F., eds., *Proceedings of 1st International Conference on Principles and Practice of Constraint Programming, CP95*, 276–291. Springer-Verlag.

Jeavons, P.; Cohen, D.; and Gyssens, M. 1996. A test for tractability. In Freuder, E., ed., *Proceedings of 2nd International Conference on Principles and Practice of Constraint Programming, CP96*, 267–281. Springer-Verlag.

Jeavons, P.; Cohen, D.; and Gyssens, M. 1997. Closure properties of constraints. *Journal of the ACM* 44(4):527–48.

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

Kolaitis, P., and Vardi, M. 1998. Conjunctive-query containment and constraint satisfaction. In *Proc. 17th ACM Symp. on Principles of Database Systems*, 205–213. Full version at `http://www.cs.rice.edu/~vardi/papers`.

Mackworth, A., and Freuder, E. 1993. The complexity of constraint satisfaction revisited. *Artificial Intelligence* 59(1-2):57–62.

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

van Beek, P., and Dechter, R. 1997. Constraint tightness and looseness versus local and global consistency. *Journal of the ACM* 44(4):549–566.

van Beek, P. 1994. On the inherent tightness of local consistency in constraint networks. In *Proc. of National Conference on Artificial Intelligence (AAAI-94)*, 368–373.