Annals of Mathematics and Artificial Intelligence 24 (1998) 51–67

Constraints and universal algebra

Peter Jeavons [a], David Cohen [a] and Justin Pearson [b]

[a] *Department of Computer Science, Royal Holloway, University of London, UK*
E-mail: p.jeavons@dcs.rhbnc.ac.uk
[b] *Department of Computer Systems, Uppsala University, S-751 05 Uppsala, Sweden,*
E-mail: justin@docs.uu.se

In this paper we explore the links between constraint satisfaction problems and *universal algebra*. We show that a constraint satisfaction problem instance can be viewed as a pair of relational structures, and the solutions to the problem are then the structure preserving mappings between these two relational structures. We give a number of examples to illustrate how this framework can be used to express a wide variety of combinatorial problems, many of which are not generally considered as constraint satisfaction problems. We also show that certain key aspects of the mathematical structure of constraint satisfaction problems can be precisely described in terms of the notion of a *Galois connection,* which is a standard notion of universal algebra. Using this result, we obtain an algebraic characterisation of the property of minimality in a constraint satisfaction problem. We also obtain a similar algebraic criterion for determining whether or not a given set of solutions can be expressed by a constraint satisfaction problem with a given structure, or a given set of allowed constraint types.

1. Introduction

The constraint satisfaction problem provides a framework in which it is possible to express, in a natural way, many combinatorial problems encountered in artificial intelligence and elsewhere. The aim in a constraint satisfaction problem is to find an assignment of values to a given set of variables subject to constraints on the values which can be assigned simultaneously to certain specified subsets of variables.

The mathematical framework used to describe constraint satisfaction problems has strong links with several other areas of computer science and mathematics.

In a previous paper [5] we explored one example of this by describing and using the close links between constraint satisfaction problems and relational databases. Relational database theory provides a very powerful and convenient terminology for describing operations on relations, and certain aspects of their structure, and this terminology is very useful in the description and study of constraint satisfaction problems. Furthermore, many of the computational tasks undertaken in the processing and solution of constraint satisfaction problems also arise in the context of database management. We believe that a greater interchange of ideas between these two fields could be of considerable benefit to both [2,5], and therefore these links should receive more attention than they currently do.

In the present paper we explore a different set of links, which we believe will prove to be equally important, the links between constraint satisfaction problems and *universal algebra*. This branch of mathematics is concerned with the study of the structure of general mathematical objects such as abstract algebras, lattices, and relational structures, and with the operations that can be defined on such objects.

We shall show that a constraint satisfaction problem instance can be viewed as a pair of relational structures, and the solutions to the problem are then the structure preserving mappings between these two relational structures.

One advantage of viewing constraint satisfaction problems in this way is that it suggests a number of algebraic techniques for analysing the properties of a given problem class. For example, the nature of the constraints which occur in a problem class can be described by specifying the algebraic operations under which they are invariant. This has led to a novel approach to the study of tractability in constraint satisfaction problems which focuses on algebraic properties of constraints [9–11]. This approach has led to a number of new insights into the nature of tractable constraint types. In particular, we have established that any collection of tractable constraints over a finite domain, must all be invariant under a pointwise operation [10,11]. This result has transformed the search for new tractable constraint types into a search for possible algebraic invariance properties.

Another advantage of an algebraic viewpoint is that it allows the powerful structural results obtained in the field of universal algebra to be applied to the analysis of constraint satisfaction problems. For example, a theorem about algebras obtained by Baker and Pixley in 1975 [1] provides a precise algebraic characterisation of the constraint types for which local consistency is sufficient to ensure global consistency [8].

As another example, we shall show below that certain key aspects of the mathematical structure of constraint satisfaction problems can be precisely described in terms of the notion of a *Galois connection*, which is a standard notion of universal algebra. In particular, we show that it is possible using this algebraic framework to give a precise description of the notion of "expressive power" for constraints. This includes the expressive power of a fixed constraint structure (section 3) and the expressive power of a fixed collection of constraint types (section 4). In both cases we show that the results can be used to determine whether or not a particular relation can be expressed in a given framework. The question of determining whether a given relation can be expressed using a fixed collection of constraint types is a significant theoretical question which is related to the design of constraint programming languages, and the answer we provide here uses some very general results from universal algebra to provide a surprisingly simple approach (theorem 4.6).

The paper is organised as follows. In section 2 we give the basic definitions, and establish the fundamental connection between constraint satisfaction problems and relational structures. In section 3 we consider the class of constraint satisfaction problems with a given constraint structure, that is, a fixed constraint hypergraph. We investigate the relationship between problem instances and sets of solutions, and show that this relationship can be viewed as a Galois connection. Hence we obtain an algebraic characterisation of the property of minimality, and an algebraic criterion for determining whether or not a given set of solutions can be expressed by a constraint problem of a certain arity. In section 4 we consider the class of constraint satisfaction problems with arbitrary constraint hypergraphs, but with a given collection of constraint types. We show how algebraic properties known as *polymorphisms* can be used to classify different constraint types, and we indicate how the relationship between constraint types and polymorphisms can also be viewed as an example of a Galois connection. Finally, we show that the “expressive power” of a set of constraint types, and the complexity of the corresponding class of problems, can be determined from the mappings in this Galois connection.

2. Definitions

2.1. The constraint satisfaction problem

The fundamental mathematical structure required to describe constraints, and constraint satisfaction problems, is the *relation*, which is defined as follows.

Definition 2.1. For any set D, and any natural number n, we denote the set of all n-tuples of elements of D by D^n. A subset of D^n is called an ‘*n-ary relation*’ over D.

For any tuple $t \in D^n$, and any i in the range 1 to n, we denote the value in the ith coordinate position of t by $t[i]$. The tuple t will be written in the form $\langle t[1], t[2], \ldots, t[n]\rangle$.

The following relations will be of special interest in this paper:

Definition 2.2. For any set D we define the following binary relations over D:

- equality: $\square_D = \{\langle d, d'\rangle \in D^2 \mid d = d'\}$,
- disequality: $\neq_D = \{\langle d, d'\rangle \in D^2 \mid d \neq d'\}$.

The definition of a ‘constraint satisfaction problem’ varies slightly between authors [11, 13,14,16], but the following is a fairly standard version of the definition.

Definition 2.3. An instance of a *constraint satisfaction problem* is a triple $\langle V, D, \mathcal{C}\rangle$, where:

- V is a set of *variables*,
- D is a domain of *values*,
- $\mathcal{C}$ is a set of *constraints*, $\{C_1, C_2, \ldots, C_q\}$.
  Each constraint $C_i \in \mathcal{C}$ is a pair $\langle s_i, R_i\rangle$, where:
  - s_i is a tuple of variables of length m_i, called the ‘*constraint scope*’, and
  - R_i is an m_i-ary relation over D, called the ‘*constraint relation*’.

For each constraint, $\langle s_i, R_i\rangle$, the tuples of R_i indicate the allowed combinations of simultaneous values for the variables in s_i. The length of the tuples in R_i is called the 'arity' of the constraint. In particular, unary constraints specify the allowed values for a single variable, and binary constraints specify the allowed combinations of values for a pair of variables.

A *solution* to a constraint satisfaction problem instance is a function[1], f, from the set of variables, V, to the domain of values, D, such that for each constraint $\langle s_i, R_i\rangle$, with $s_i = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_m}\rangle$, the tuple $\langle f(v_{i_1}), f(v_{i_2}), \ldots, f(v_{i_m})\rangle$ is a member of R_i. Each instance of a constraint satisfaction problem is associated with a set of solutions, and can be said to 'represent' or 'express' this set of solutions.

Example 2.4. An instance of the standard propositional SATISFIABILITY problem [4,17] is specified by giving a formula in propositional logic, and asking whether there are values for the variables which make the formula true.

For example, consider the formula

$$(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_3 \vee \overline{x}_4 \vee x_1) \wedge (\overline{x}_3 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_3).$$

The problem of finding a satisfying truth assignment for this formula can be formulated as an instance of the constraint satisfaction problem in a number of ways. Perhaps the most straightforward is to construct the instance $\mathcal{P}$ with

- set of variables $V = \{x_1, x_2, x_3, x_4\}$,
- set of values $D = \{0, 1\}$, corresponding to True and False,
- set of constraints $\{C_1, C_2, C_3, C_4, C_5\}$, where:
  - $C_1 = \langle\langle x_1, x_2, x_3, x_4\rangle, D^4 \setminus \langle 0,0,0,0\rangle\rangle$,
  - $C_2 = \langle\langle x_1, x_2, x_3\rangle, D^3 \setminus \langle 1,1,0\rangle\rangle$,
  - $C_3 = \langle\langle x_3, x_4, x_1\rangle, D^3 \setminus \langle 1,1,0\rangle\rangle$,
  - $C_4 = \langle\langle x_3, x_2, x_4\rangle, D^3 \setminus \langle 1,1,0\rangle\rangle$,
  - $C_5 = \langle\langle x_1, x_3\rangle, D^2 \setminus \langle 1,1\rangle\rangle$.

A simple calculation reveals that there are 6 solutions to this problem instance, which can be tabulated as follows:

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	0
0	1	0	0
0	1	0	1
1	0	0	0
1	0	0	1

[1] Some authors use the term solution to refer to the *image* of such a function on some fixed ordering of the variables, rather than the function itself [13]. These two notions are clearly very closely related, but for our purposes it is more convenient to regard the function itself as the solution (see proposition 2.9).

2.2. Relational structures and homomorphisms

In order to describe constraint satisfaction problems in algebraic terms, we will make extensive use of the standard algebraic notion of a 'relational structure' [3,15].

Definition 2.5. A *'relational structure'* is a tuple $\langle V, E_1, E_2, \ldots, E_k \rangle$ consisting of a non-empty set, V, called the *'universe'* of the relational structure, and a list, $E_1, E_2, \ldots, E_k$, of relations over V.

Example 2.6. A (directed) graph is a relational structure in which the universe is a set, V, of vertices, and there is a single binary relation, E, specifying which vertices are adjacent.

A complete graph on n vertices, denoted K_n, corresponds to a relational structure $\langle V, \neq_V \rangle$, where V is a set of cardinality n, and $\neq_V$ is the disequality relation over V defined in example 2.2.

Definition 2.7. The *'rank function'* of a relational structure $\langle V, E_1, E_2, \ldots, E_k \rangle$, is a function ρ from $\{1, 2, \ldots, k\}$ to the set of non-negative integers, such that for all $i \in \{1, 2, \ldots, k\}$, $\rho(i)$ is the arity of E_i.

A relational structure Σ is *'similar'* to a relational structure Σ' if they have identical rank functions.

Definition 2.8. Let $\Sigma = \langle V, E_1, E_2, \ldots, E_k \rangle$ and $\Sigma' = \langle V', E'_1, E'_2, \ldots, E'_k \rangle$ be two similar relational structures, and let ρ be their common rank function.

A *'homomorphism'* from Σ to Σ' is a function $h : V \to V'$ such that, for all $i \in \{1, 2, \ldots, k\}$,

$$\langle v_1, v_2, \ldots, v_{\rho(i)} \rangle \in E_i \Longrightarrow \langle h(v_1), h(v_2), \ldots, h(v_{\rho(i)}) \rangle \in E'_i.$$

The set of all homomorphisms from Σ to Σ' is denoted $\mathrm{Hom}(\Sigma, \Sigma')$.

The fundamental connection we wish to explore in this paper is that the solutions to any constraint satisfaction problem instance can be viewed as homomorphisms between a fixed pair of relational structures, as the next result indicates.

Proposition 2.9. For any constraint satisfaction problem instance $\mathcal{P} = \langle V, D, \mathcal{C} \rangle$, with $\mathcal{C} = \{\langle s_1, R_1 \rangle, \langle s_2, R_2 \rangle, \ldots, \langle s_q, R_q \rangle\}$, the set of solutions to $\mathcal{P}$ equals $\mathrm{Hom}(\Sigma, \Sigma')$, where $\Sigma = \langle V, \{s_1\}, \{s_2\}, \ldots, \{s_q\} \rangle$ and $\Sigma' = \langle D, R_1, R_2, \ldots, R_q \rangle$.

Example 2.10. Reconsider the constraint satisfaction problem instance $\mathcal{P}$, with set of variables $V = \{x_1, x_2, x_3, x_4\}$ and set of values $D = \{0, 1\}$, defined in example 2.4. It follows from proposition 2.9 that the six solutions to $\mathcal{P}$, listed in example 2.4 are precisely the homomorphisms from Σ to Σ', where:

- $\Sigma = \langle V, \{\langle x_1, x_2, x_3, x_4 \rangle\}, \{\langle x_1, x_2, x_3 \rangle\}, \{\langle x_3, x_4, x_1 \rangle\}, \{\langle x_3, x_2, x_4 \rangle\}, \{\langle x_1, x_3 \rangle\}\rangle$,

- $\Sigma' = \langle D, R_1, R_2, R_2, R_2, R_3 \rangle$, where:
  - $R_1 = \{0,1\}^4 \setminus \langle 0,0,0,0 \rangle$,
  - $R_2 = \{0,1\}^3 \setminus \langle 1,1,0 \rangle$,
  - $R_3 = \{0,1\}^2 \setminus \langle 1,1 \rangle$.

Conversely, given any pair of similar relational structures, there is a corresponding constraint satisfaction problem instance whose solutions are precisely the homomorphisms between the two structures, as the next result indicates.

Proposition 2.11. For any pair of similar relational structures, Σ and Σ', with $\Sigma = \langle V, E_1, E_2, \ldots, E_k \rangle$ and $\Sigma' = \langle D, R_1, R_2, \ldots, R_k \rangle$, the set $\mathrm{Hom}(\Sigma, \Sigma')$ is equal to the set of solutions to the constraint satisfaction problem instance $\langle V, D, \mathcal{C} \rangle$ where $\mathcal{C} = \bigcup_{i=1}^{k} \{\langle s, R_i \rangle \mid s \in E_i\}$.

Example 2.12. Consider the relational structures Σ and Σ', where:

- $\Sigma = \langle V, \{\langle x_1, x_2, x_3, x_4 \rangle\}, \{\langle x_1, x_2, x_3 \rangle, \langle x_3, x_4, x_1 \rangle, \langle x_3, x_2, x_4 \rangle\}, \{\langle x_1, x_3 \rangle\} \rangle$, where:
  - $V = \{x_1, x_2, x_3, x_4\}$,
- $\Sigma' = \langle \{0,1\}, R_1, R_2, R_3 \rangle$, where:
  - $R_1 = \{0,1\}^4 \setminus \langle 0,0,0,0 \rangle$,
  - $R_2 = \{0,1\}^3 \setminus \langle 1,1,0 \rangle$,
  - $R_3 = \{0,1\}^2 \setminus \langle 1,1 \rangle$.

Note that these relational structures are not equal to the relational structures defined in example 2.10 (for example, they each contain 3 relations rather than 5). However, it follows from proposition 2.11 that the set $\mathrm{Hom}(\Sigma, \Sigma')$ is again equal to the set of solutions to the constraint satisfaction problem instance $\mathcal{P}$, defined in example 2.4.

In view of these results, we shall refer to any pair of similar relational structures, $\langle \Sigma, \Sigma' \rangle$, as an instance of the 'generalised constraint satisfaction problem' (GCP), and we shall regard the set of homomorphisms $\mathrm{Hom}(\Sigma, \Sigma')$ as the set of solutions to this problem instance.

Definition 2.13. An instance of a *generalised constraint satisfaction problem* is a pair $\langle \Sigma, \Sigma' \rangle$, where Σ and Σ' are similar relational structures.

A *solution* to $\langle \Sigma, \Sigma' \rangle$ is a homomorphism from Σ to Σ'.

Comparing definition 2.13 with the more usual definition 2.3 indicates how the use of standard algebraic terminology streamlines the definition.

Notice that in any problem instance $\langle \Sigma, \Sigma' \rangle$, the first component, Σ, specifies the structure of the constraints (which variables constrain which others), sometimes called

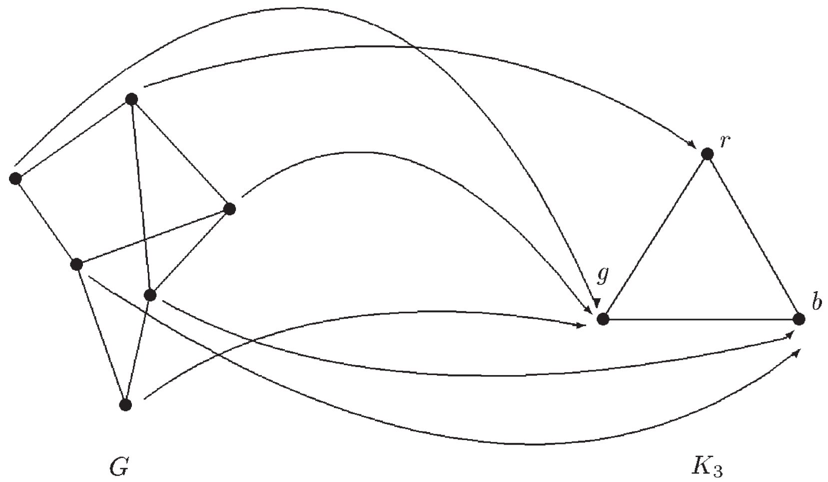

Figure 1. A solution to a GRAPH COLORABILITY problem instance.

the 'constraint (hyper)graph', whilst the second component, Σ', specifies the constraint relations.

The general constraint satisfaction framework described in definition 2.13 allows many standard combinatorial problems to be expressed very simply as instances of GCP, as the following examples indicate. Note that many of these examples concern problems which are not usually viewed as constraint satisfaction problems.

Example 2.14 (GRAPH COLORABILITY). An instance of the GRAPH COLORABILITY problem [4,17] consists of a graph G and an integer q. The question is whether the vertices of G can be labelled with q colours in such a way that adjacent vertices are labelled with different colours.

This can be expressed as the GCP instance $\langle G, K_q \rangle$, where K_q is a complete graph on q vertices, as defined in example 2.6.

For example, figure 1 indicates a homomorphism from a graph G to the complete graph K_3, which corresponds to a 3-colouring of G.

Example 2.15 (CLIQUE). An instance of the CLIQUE problem [4,17] consists of a graph G and an integer q. The question is whether G contains a subgraph of q vertices which is a clique (that is, isomorphic to a complete graph K_q).

Assuming that G contains no 'loops' (in other words, no vertex is adjacent to itself), this can be expressed as the GCP instance $\langle K_q, G \rangle$.

For example, figure 2 indicates a homomorphism from the complete graph K_3 to a graph G, which corresponds to finding a 3-clique in G.

Example 2.16 (HAMILTONIAN CIRCUIT). An instance of the HAMILTONIAN CIRCUIT problem [4,17] consists of a graph $G = \langle V, E \rangle$. The question is whether there is a

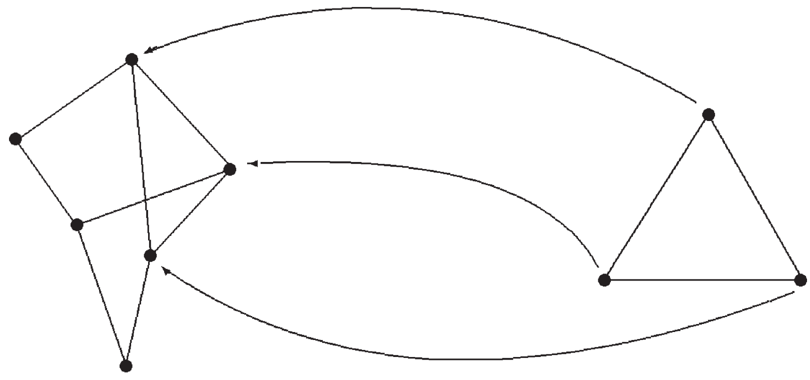

Figure 2. A solution to a CLIQUE problem instance.

cyclic ordering of V such that every pair of successive nodes in the ordering is adjacent in G.

This can be expressed as the GCP instance $\langle\langle V, C_V, \neq_V\rangle, \langle V, E, \neq_V\rangle\rangle$, where C_V is an arbitrary cyclic permutation on V and $\neq_V$ is the disequality relation over V defined above. (The presence of the relation $\neq_V$ in both relational structures simply ensures that any solution must be injective.)

Example 2.17 (BANDWIDTH). An instance of the BANDWIDTH problem [4] consists of a graph $G = \langle V, E\rangle$ where $V = \{v_1, v_2, \ldots, v_n\}$, and a positive integer k. The question is whether there is a linear ordering of V such that adjacent nodes in the graph are at most k positions apart in the ordering.

This can be expressed as the GCP instance $\langle\langle V, E, \neq_V\rangle, \langle V, B_k, \neq_V\rangle\rangle$, where $B_k = \{\langle v_i, v_j\rangle \in V^2 \mid |i - j| \leqslant k\}$ and $\neq_V$ is the disequality relation over V defined above.

Example 2.18 (k-DIMENSIONAL MATCHING). An instance of the k-DIMENSIONAL MATCHING problem [4] consists of a relation M of arity k over a set V. The question is whether there is a subset $M' \subseteq M$ such that $|M'| = |V|$ and no two elements of M' agree in any coordinate position.

This can be expressed as the GCP instance $\langle\langle V, \neq_V\rangle, \langle M, \tilde{\neq}_M\rangle\rangle$, where

$$\tilde{\neq}_M = \bigl\{\bigl\langle\langle v_1, v_2, \ldots, v_k\rangle, \langle v_1', v_2', \ldots, v_k'\rangle\bigr\rangle \in M^2 \mid v_i \neq v_i',\ i = 1, 2, \ldots, k\bigr\}.$$

All of the above problems are known to be NP-complete [4,17], but the next two examples show that it is not just NP-complete problems which can be expressed in this framework.

Example 2.19 (GRAPH ISOMORPHISM). An instance of the GRAPH ISOMORPHISM problem [4,17] consists of two graphs $G = \langle V, E\rangle$ and $G' = \langle V', E'\rangle$ with $|V| = |V'|$.

The question is whether there is a bijection between the vertices such that adjacent vertices in G are mapped to adjacent vertices in G', and non-adjacent vertices in G are mapped to non-adjacent vertices in G'.

This can be expressed as the GCP instance $\langle\langle V, E, \bar{E}\rangle, \langle V', E', \bar{E}'\rangle\rangle$, where $\bar{E} = \neq_V \setminus E$ and $\bar{E}' = \neq_{V'} \setminus E'$.

Example 2.20 (GRAPH UNREACHABILITY). An instance of the GRAPH UNREACHABILITY problem [17] consists of a graph $G = \langle V, E\rangle$ and a pair of vertices, $v, w \in V$. The question is whether there is no path in G connecting v to w.

This can be expressed as the GCP instance

$$\big\langle\langle V, E, \{\langle v\rangle\}, \{\langle w\rangle\}\big\rangle, \big\langle\{0,1\}, \square_{\{0,1\}}, \{\langle 0\rangle\}, \{\langle 1\rangle\}\big\rangle\big\rangle.$$

3. Fixed constraint hypergraph

In this section, we shall consider the collection of all instances of the generalised constraint satisfaction problem with a fixed constraint hypergraph, and a fixed domain of values, but with differing constraint relations.

To do this, we choose a fixed relational structure, Σ_0, with universe V and rank function ρ, and a fixed set D. We then consider all GCP instances $\langle\Sigma_0, \Sigma\rangle$, where Σ varies over all relational structures with universe D and rank function ρ.

To obtain a partial ordering on this set of problem instances we note that similar relational structures with the same universe can be partially ordered by inclusion on the corresponding relations, as follows.

Definition 3.1. If $\Sigma = \langle D, E_1, E_2, \ldots, E_k\rangle$ and $\Sigma' = \langle D, E'_1, E'_2, \ldots, E'_k\rangle$ are similar relational structures, with $E_i \supseteq E'_i$ for all $i \in \{1, 2, \ldots, k\}$, then we shall say[2] that $\Sigma \leqslant \Sigma'$.

Using this partial order, the set of all relational structures with universe D and rank function ρ forms a *lattice* [3,15], which we shall call $\Delta_{D,\rho}$.

Now, each problem instance $\langle\Sigma_0, \Sigma\rangle$ has an associated set of solutions, which is a set of mappings from V to D. Furthermore, the set of all sets of mappings from V to D, denoted $\wp(D^V)$, can be partially ordered by inclusion, in the standard way, and also forms a lattice.

Given any fixed relational structure, Σ_0, with universe V and rank function ρ we can define a pair of mappings between $\Delta_{D,\rho}$ and $\wp(D^V)$, as follows.

Definition 3.2. Let $\Sigma_0 = \langle V, E_1, E_2, \ldots, E_k\rangle$ be a relational structure, with universe V and rank function ρ.

We define two mappings, $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$ as follows:

[2] Note that the ordering defined here is reversed from the standard inclusion ordering, for convenience later.

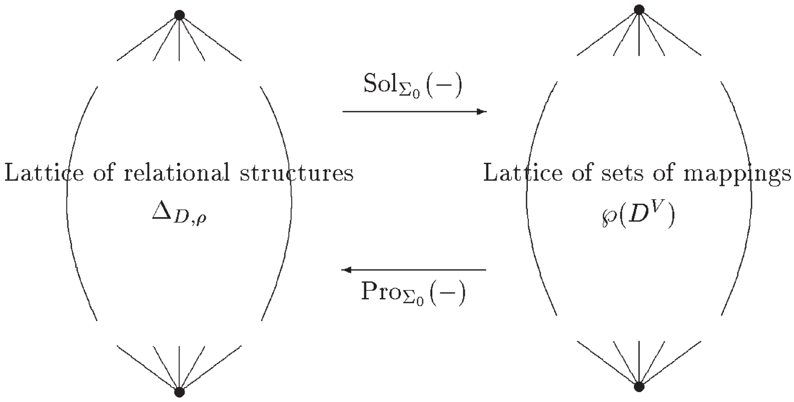

Figure 3. The Galois connection between $\Delta_{D,\rho}$ and $\wp(D^V)$.

- For any relational structure Σ with rank function ρ, define $\mathrm{Sol}_{\Sigma_0}(\Sigma)$ as follows:

$$\mathrm{Sol}_{\Sigma_0}(\Sigma) = \mathrm{Hom}(\Sigma_0, \Sigma).$$

- For any set of mappings, M, from V to some set D, define $\mathrm{Pro}_{\Sigma_0}(M)$ as follows:

$$\mathrm{Pro}_{\Sigma_0}(M) = \langle D, R_1, R_2, \ldots, R_k \rangle,$$

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

$$R_i = \bigcup_{\langle v_1, v_2, \ldots, v_{\rho(i)} \rangle \in E_i} \bigl\{ \bigl\langle m(v_1), m(v_2), \ldots, m(v_{\rho(i)}) \bigr\rangle \mid m \in M \bigr\}.$$

This pair of mappings is illustrated in figure 3. The mapping $\mathrm{Sol}_{\Sigma_0}(-)$ takes each relational structure Σ to the set of solutions to $\langle \Sigma_0, \Sigma \rangle$. Conversely, the mapping $\mathrm{Pro}_{\Sigma_0}(-)$ takes each set of mappings to a relational structure Σ such that the problem instance $\langle \Sigma_0, \Sigma \rangle$ 'expresses' this set of mappings as closely as possible, given the fixed choice of constraint hypergraph.

From an algebraic point of view, the significant fact about this pair of mappings is that they are order-reversing, as shown in the next result.

Proposition 3.3. For any relational structure $\Sigma_0 = \langle V, E_1, E_2, \ldots, E_q \rangle$ with rank function ρ, the mappings $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$ have the following properties:

- For any two relational structures Σ_1 and Σ_2, that have a common universe, and rank function ρ, if $\Sigma_1 \leqslant \Sigma_2$, then $\mathrm{Sol}_{\Sigma_0}(\Sigma_1) \geqslant \mathrm{Sol}_{\Sigma_0}(\Sigma_2)$.
- For any two sets of mappings, M_1 and M_2, in $\wp(D^V)$, if $M_1 \leqslant M_2$, then $\mathrm{Pro}_{\Sigma_0}(M_1) \geqslant \mathrm{Pro}_{\Sigma_0}(M_2)$.
- For any Σ with rank function ρ, $\mathrm{Pro}_{\Sigma_0}(\mathrm{Sol}_{\Sigma_0}(\Sigma)) \geqslant \Sigma$.
- For any M in $\wp(D^V)$, $\mathrm{Sol}_{\Sigma_0}(\mathrm{Pro}_{\Sigma_0}(M)) \geqslant M$.

This result states that the mappings $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$ constitute a *Galois connection* [3,15] between $\Delta_{D,\rho}$ and $\wp(D^V)$, for any set D.

As with any Galois connection [3,15], this implies that the composite operation $\mathrm{Pro}_{\Sigma_0}(\mathrm{Sol}_{\Sigma_0}(-))$ is a closure operation on $\Delta_{D,\rho}$, and the elements which are fixed by this operation form a lattice.

For certain choices of Σ_0 these elements are of particular interest. For example, in Montanari's original 1974 paper on constraints [16], he defines the notion of a 'minimal constraint network'. This is a binary constraint satisfaction problem in which the constraint between every pair of variables is as tight as possible. In other words, every pair of values allowed by every constraint can be extended to a complete solution. Montanari calls the problem of deriving the (unique) minimal constraint network with the same solutions as a given problem instance the "central problem" in many practical applications [16].

Proposition 3.4. Let $K_n = \langle V, \{e_1, e_2, \ldots\}\rangle$ be a complete graph with n vertices, and set $\Sigma_{Kn} = \langle V, \{e_1\}, \{e_2\}, \ldots\rangle$.

For any GCP instance $\mathcal{P} = \langle \Sigma_{Kn}, \Sigma'\rangle$, the unique *minimal* binary constraint satisfaction problem (as defined in [16]) with the same solutions as $\mathcal{P}$ is given by

$$\left\langle \Sigma_{Kn}, \mathrm{Pro}_{\Sigma_{Kn}}\left(\mathrm{Sol}_{\Sigma_{Kn}}(\Sigma)\right)\right\rangle.$$

Hence $\mathcal{P}$ is a minimal binary constraint satisfaction problem if and only if $\mathrm{Pro}_{\Sigma_{Kn}}(\mathrm{Sol}_{\Sigma_{Kn}}(\Sigma)) = \Sigma$.

Similarly, the composite operation $\mathrm{Sol}_{\Sigma_0}(\mathrm{Pro}_{\Sigma_0}(-))$ is a closure operation on $\wp(D^V)$, and the elements which are fixed by this operation also form a lattice.

Proposition 3.5. A set of mappings M in $\wp(D^V)$ is the set of solutions to some constraint satisfaction problem instance with constraint hypergraph Σ_0 if and only if

$$\mathrm{Sol}_{\Sigma_0}\left(\mathrm{Pro}_{\Sigma_0}(M)\right) = M.$$

For some choices of Σ_0 this result is of particular interest.

Corollary 3.6. Let $K_n = \langle V, \{e_1, e_2, \ldots\}\rangle$ be a complete graph with n vertices, and set $\Sigma_{Kn} = \langle V, \{e_1\}, \{e_2\}, \ldots\rangle$.

A set of mappings M in $\wp(D^V)$ is the set of solutions to some binary constraint satisfaction problem instance if and only if $\mathrm{Sol}_{\Sigma_{Kn}}(\mathrm{Pro}_{\Sigma_{Kn}}(M)) = M$.

Similar results for other problem arities can be obtained by choosing a Σ_0 which corresponds to a complete hypergraph of the appropriate degree.

Example 3.7. Let S be the set of solutions to the GCP instance $\langle \Sigma, \Sigma'\rangle$ defined in example 2.10. The set S contains 6 mappings from $V = \{x_1, x_2, x_3, x_4\}$ to $D = \{0, 1\}$, as indicated in example 2.4.

In this instance we have

$$\Sigma_{K_n} = \bigl\langle V, \bigl\{\langle x_1, x_2\rangle\bigr\}, \bigl\{\langle x_1, x_3\rangle\bigr\}, \bigl\{\langle x_1, x_4\rangle\bigr\}, \bigl\{\langle x_2, x_3\rangle\bigr\}, \bigl\{\langle x_2, x_4\rangle\bigr\}, \bigl\{\langle x_3, x_4\rangle\bigr\}\bigr\rangle,$$

giving

$$\begin{aligned}\mathrm{Pro}_{\Sigma_{K_n}}(S) = \bigl\langle V, &\bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\bigr\}, \bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\bigr\},\\ &\bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle\bigr\}, \bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\bigr\},\\ &\bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle\bigr\}, \bigl\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\bigr\}\bigr\rangle,\end{aligned}$$

which then gives

$$\mathrm{Sol}_{\Sigma_{K_n}}\bigl(\mathrm{Pro}_{\Sigma_{K_n}}(S)\bigr) = S \cup \bigl\{\langle 0,0,0,0\rangle\bigr\} \neq S.$$

Hence, by corollary 3.6, S cannot be expressed as the set of solutions to *any* binary constraint satisfaction problem instance.

A similar calculation shows that S cannot be expressed as the set of solutions to any ternary constraint satisfaction problem instance (but see example 4.8).

4. Fixed constraint relations

In this section, we shall consider the collection of all instances of the generalised constraint satisfaction problem with a fixed domain of values, and a fixed set of possible constraint relations, but with differing constraint hypergraphs.

Throughout this section, we shall assume that the domain of values for all the problems we consider is some fixed set D. For any set, Γ, of relations over D, the collection of all instances, $\langle \Sigma, \Sigma' \rangle$, of the generalised constraint satisfaction problem where the relations of Σ' are all elements of Γ, will be denoted GCP(Γ).

Once again, the notion of a Galois Connection turns out to be of fundamental importance for studying the class of problems GCP(Γ). In this case we shall construct a mapping from each set of relations Γ to an associated set of operations on D, which are called the *polymorphisms*[3] of Γ, as described below. We also construct a second mapping from sets of operations to sets of relations, and show that these two mappings form a Galois connection.

First note that the set of all possible sets of relations over D is partially ordered by inclusion in the standard way, and forms an infinite lattice, which we shall call Λ_D.

An *operation* on a set D is a function from D^m to D, for some positive integer m, which is called the *arity* of the operation. The set of all sets of operations on D is again partially ordered by inclusion in the standard way, and forms an infinite lattice, which we shall call Ω_D.

[3] In previous papers [9–11] we have used the term ‘closure operation’ instead of ‘polymorphism’, but we introduce the term polymorphism here for consistency with the literature of universal algebra, and to avoid confusion with the notion of closure arising from the Galois connection.

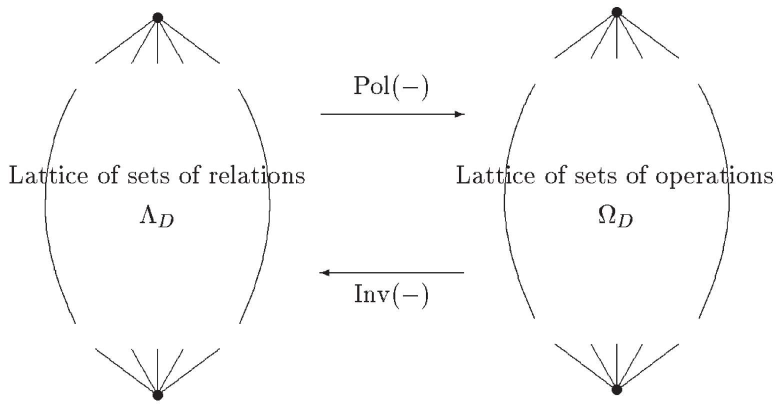

Figure 4. The Galois connection between Λ_D and Ω_D.

Definition 4.1. Let R be a relation over D. An operation f of arity n on D is a *polymorphism* of R if

$$\forall t_1, t_2, \ldots, t_n \in R \quad f(t_1, t_2, \ldots, t_n) \in R,$$

where the operation f is applied componentwise.

If f is a polymorphism of R, then R is said to be *invariant* for f.

We now define a pair of order-preserving mappings between Λ_D and Ω_D, as follows.

Definition 4.2. We define two mappings, Pol(−) and Inv(−) as follows:

- For any set of relations Γ over D, define Pol(Γ) as follows:

$$\mathrm{Pol}(\Gamma) = \{ f \mid \forall R \in \Gamma,\ f \text{ is a polymorphism of } R \}.$$

- For any set of operations, O, on D, define Inv(O) as follows:

$$\mathrm{Inv}(O) = \{ R \mid \forall f \in O,\ R \text{ is invariant for } f \}.$$

These mappings are illustrated in figure 4. From an algebraic point of view, the significant fact about this pair of mappings is that they are order-reversing, as shown in the next result.

Proposition 4.3. The mappings Pol(−) and Inv(−) have the following properties:

- For any two sets of relations Γ_1 and Γ_2, over D, if $\Gamma_1 \subseteq \Gamma_2$, then Pol($\Gamma_1$) $\supseteq$ Pol(Γ_2).
- For any two sets of operations, O_1 and O_2, on D, if $O_1 \subseteq O_2$, then Inv(O_1) $\supseteq$ Inv(O_2).
- For any set of relations, Γ over D, Inv(Pol(Γ)) $\supseteq \Gamma$.
- For any set of operations O on D, Pol(Inv(O)) $\supseteq O$.

This result states that the mappings Pol($-$) and Inv($-$) constitute a Galois connection between Λ_D and $\mathbf{\Omega}_D$. This particular Galois connection is well-known in universal algebra, and has been extensively studied [6,15,19].

As with any Galois connection, the composite operation Pol(Inv($-$)) is a closure operation on $\mathbf{\Omega}_D$, and the elements which are fixed by this operation form a lattice. These fixed elements are sets of operations with particular algebraic properties, which have been investigated in universal algebra under the name of 'clones'.

Definition 4.4 [19]. A set of operations O on a set D is called a *clone* if

- every projection operation on D is an element of O, and
- the composition of any elements in O is an element of O.

Proposition 4.5. A set of operations O on D is a *clone* if and only if Pol(Inv(O)) $= O$.

Similarly, the composite operation Inv(Pol($-$)) is a closure operation on Λ_D, and the elements which are fixed by this operation also form a lattice. These fixed elements are in one-to-one correspondence with the clones described above.

The significance of the operation Inv(Pol($-$)), from the point of view of the general constraint satisfaction problem, is shown by the next result. This result states, in effect, that a relation belongs to Inv(Pol(Γ)) if and only if it is equal to a projection of the set of solutions to some problem in GCP(Γ). In other words, the set of relations Inv(Pol(Γ)) contains precisely the relations that can be 'expressed' by problem instances in GCP(Γ).

Theorem 4.6. Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of variables and let M be a set of mappings from V to a finite set D.

For any fixed set of relations Γ over D, there exists some constraint satisfaction problem instance in GCP($\Gamma \cup \Box_D$) with solution set S such that $S|_V = M$ if and only if

$$\bigl\{\bigl\langle f(v_1), f(v_2), \ldots, f(v_n)\bigr\rangle \mid f \in M\bigr\} \in \mathrm{Inv}\bigl(\mathrm{Pol}(\Gamma)\bigr).$$

Proof. It is noted in [6] that a relation R belongs to Inv(Pol(Γ)) if and only if there exists a formula $\phi(x_1, x_2, \ldots, x_n)$ with free variables $x_1, x_2, \ldots, x_n$, in a fragment of first order logic containing only binary equality, conjunction, and existential quantification, together with an m-ary predicate symbol for each m-ary relation in Γ, such that

$$R = \bigl\{\langle d_1, d_2, \ldots, d_n\rangle \in D^n \mid \Gamma \models \phi(d_1, d_2, \ldots, d_n)\bigr\}.$$

Furthermore, each such formula corresponds to a constraint satisfaction problem instance in GCP($\Gamma \cup \Box_D$) [14]. (Variables in the formula which are existentially quantified give rise to variables in the corresponding constraint satisfaction problem instance which are "hidden", that is, whose values are ignored.) $\Box$

This is a very powerful result, which can be viewed in the following way: if we think of Γ as a 'language' of possible constraint relations which can be imposed explicitly on collections of variables, then Inv(Pol(Γ)) is precisely the set of relations which can be imposed either explicitly or implicitly on some subset of variables using this language [12].

For some choices of Γ this result is of particular interest.

Corollary 4.7. Let $\Gamma^{(r)}$ be the set of all relations over D of arity r.

For any set of mappings $M \subseteq D^V$, there exists some constraint satisfaction problem instance where the constraints have arity r with solution set S such that $S|_V = M$ if and only if

$$\bigl\{ \bigl\langle f(v_1), f(v_2), \ldots, f(v_n) \bigr\rangle \mid f \in M \bigr\} \in \mathrm{Inv}\bigl(\mathrm{Pol}\bigl(\Gamma^{(r)}\bigr)\bigr).$$

Example 4.8. Let S be the set of solutions to the GCP instance $\langle \Sigma, \Sigma' \rangle$ defined in example 2.10. The set S contains 6 mappings from $V = \{x_1, x_2, x_3, x_4\}$ to $D = \{0, 1\}$, as indicated in example 2.4.

In this instance we have:

$$\Gamma^{(2)} = \wp\bigl(\bigl\{ \langle 0,0 \rangle, \langle 0,1 \rangle, \langle 1,0 \rangle, \langle 1,1 \rangle \bigr\}\bigr).$$

It is easily verified that every element of $\Gamma^{(2)}$ is invariant for the ternary operation μ on D which is defined as follows:

$$\mu(x, y, z) = \begin{cases} y & \text{if } y = z, \\ x & \text{otherwise.} \end{cases}$$

Hence $\mu \in \mathrm{Pol}(\Gamma^{(2)})$. But this means that $S \notin \mathrm{Inv}(\mathrm{Pol}(\Gamma^{(2)}))$, because, for example,

$$\mu\bigl(\langle 0,0,0,1 \rangle, \langle 0,0,1,0 \rangle, \langle 0,1,0,0 \rangle\bigr) = \langle 0,0,0,0 \rangle \notin S.$$

Hence, by corollary 4.7, there is no binary constraint satisfaction problem instance whose solution set (restricted to V) equals S.

This means that, over this domain, there are some relations that cannot be expressed using only binary constraints, even if we allow the use of arbitrary numbers of hidden variables.

On the other hand, it can be shown [18,19] that Pol($\Gamma^{(3)}$) contains only projection operations, hence Inv(Pol($\Gamma^{(3)}$)) contains *all* relations over D, so by corollary 4.7, there does exist at least one *ternary* constraint satisfaction problem instance whose solution set (restricted to V) equals S.

Finally, it was shown in [7] that Pol(Γ) also provides information about the computational complexity of deciding whether or not any instance of GCP(Γ) has a solution, as the next results indicate.

Theorem 4.9 [7]. Let Γ_1 and Γ_2 be finite sets of relations over D.

If $\Gamma_1 \subseteq \mathrm{Inv}(\mathrm{Pol}(\Gamma_2))$, then $\mathrm{GCP}(\Gamma_1)$ can be reduced in polynomial time to $\mathrm{GCP}(\Gamma_2)$.

Corollary 4.10. For any finite set of relations, Γ, the complexity of $\mathrm{GCP}(\Gamma)$ is determined, up to polynomial-time reductions, by $\mathrm{Pol}(\Gamma)$.

5. Conclusion

In this paper we have described how a constraint satisfaction problem can be viewed as the problem of finding a homomorphism between relational structures.

We have shown that this algebraic framework is sufficiently general to allow a wide variety of combinatorial problems to be expressed very simply as constraint satisfaction problems.

We believe that this algebraic framework provides valuable new insights into the constraint satisfaction problem because it encourages the use of algebraic techniques and tools to analyse the structure of constraint satisfaction problems, and we have given a number of examples to illustrate this process.

References

[1] K.A. Baker and A.F. Pixley, Polynomial interpolation and the chinese remainder theorem, Math. Z. 143 (1975) 165–174.

[2] D.A. Cohen, M. Gyssens and P.G. Jeavons, Derivation of constraints and database relations, in: *Proceedings 2nd International Conference on Constraint Programming – CP '96*, Boston (August 1996), Lecture Notes in Computer Science, Vol. 1118 (Springer, New York, 1996) pp. 134–148.

[3] P.M. Cohn, *Universal Algebra* (Harper & Row, 1965).

[4] M. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).

[5] M. Gyssens, P.G. Jeavons and D.A. Cohen, Decomposing constraint satisfaction problems using database techniques, Artif. Intell. 66(1) (1994) 57–89.

[6] T. Ihringer and R. Pöschel, Collapsing clones, Acta Sci. Math. (Szeged) 58 (1993) 99–113.

[7] P. Jeavons, On the algebraic structure of combinatorial problems, Theoret. Comput. Sci. 200 (1998) 185–204.

[8] P.G. Jeavons, D.A. Cohen and M.C. Cooper, Constraints, consistency and closure, Artif. Intell. 101 (1998) 251–265.

[9] P.G. Jeavons, D.A. Cohen and M. Gyssens, A unifying framework for tractable constraints, in: *Proceedings 1st International Conference on Constraint Programming – CP'95*, Cassis, France, (September 1995), Lecture Notes in Computer Science, Vol. 976 (Springer, New York, 1995) pp. 276–291.

[10] P.G. Jeavons, D.A. Cohen and M. Gyssens, A test for tractability, in: *Proceedings 2nd International Conference on Constraint Programming – CP'96*, Boston (August 1996), Lecture Notes in Computer Science, Vol. 1118 (Springer, New York, 1996) pp. 267–281.

[11] P.G. Jeavons, D.A. Cohen and M. Gyssens, Closure properties of constraints, J. ACM 44 (1997) 527–548.

[12] P.G. Jeavons, D.A. Cohen and M. Gyssens, How to determine the expressive power of constraints, Constraints (1998, to appear).

[13] P.B. Ladkin and R.D. Maddux, On binary constraint problems, J. ACM 41 (1994) 435–469.
[14] A.K. Mackworth, Consistency in networks of relations, Artif. Intell. 8 (1977) 99–118.
[15] R.N. McKenzie, G.F. McNulty and W.F. Taylor, *Algebras, Lattices and Varieties*, Vol. 1 (Wadsworth and Brooks, CA, 1987).
[16] U. Montanari, Networks of constraints: Fundamental properties and applications to picture processing, Inform. Sci. 7 (1974) 95–132.
[17] C.H. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1994).
[18] E.L. Post, *The Two-Valued Iterative Systems of Mathematical Logic*, Annals of Mathematical Studies, Vol. 5 (Princeton University Press, Princeton, 1941).
[19] A. Szendrei, *Clones in Universal Algebra*, Seminaires de Mathematiques Superieures, Vol. 99 (University of Montreal, Montreal, 1986).