ELSEVIER

Theoretical Computer Science 200 (1998) 185–204

Theoretical Computer Science

On the algebraic structure of combinatorial problems

Peter Jeavons *

Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Received March 1996; revised July 1997
Communicated by M.S. Paterson

Abstract

We describe a general algebraic formulation for a wide range of combinatorial problems including SATISFIABILITY, GRAPH COLORABILITY and GRAPH ISOMORPHISM. In this formulation each problem instance is represented by a pair of relational structures, and the solutions to a given instance are homomorphisms between these relational structures. The corresponding decision problem consists of deciding whether or not any such homomorphisms exist. We then demonstrate that the complexity of solving this decision problem is determined in many cases by simple algebraic properties of the relational structures involved. This result is used to identify tractable subproblems of SATISFIABILITY, and to provide a simple test to establish whether a given set of Boolean relations gives rise to one of these tractable subproblems.

Keywords: Complexity; Satisfiability; Relational structure; Closure; Homomorphism

1. Introduction

In this paper we show how a very wide range of combinatorial problems, including SATISFIABILITY, GRAPH COLORABILITY and GRAPH ISOMORPHISM can be expressed very naturally in the framework of universal algebra. In this framework, each instance of a problem is specified by a pair of relational structures. The first element of this pair indicates which subsets of variables in the problem are constrained in some way. The second element of the pair indicates which combinations of values are allowed for these subsets. A solution to the problem instance is a homomorphism between the two structures.

By expressing problems in this standard form, we are able to obtain general results about a wide range of combinatorial problems. In particular, we describe a partial ordering on problem classes arising from the algebraic properties of the relations involved, and show that this ordering is a refinement of the notion of reducibility. Using

* E-mail: pete@dcs.rhbnc.ac.uk.

PII S0304-3975(97)00230-2

these results we are able to establish that the complexity of certain classes of decision problems is completely determined by these simple algebraic properties of relations.

As an example of this, we consider the complexity of the GENERALIZED SATISFIABILITY problem, first described by Schaefer in 1978 [15]. Each instance of this problem is specified by a formula in propositional logic containing relation symbols corresponding to some fixed set of Boolean relations. Schaefer demonstrated that the problem of determining whether such a formula has a satisfying truth assignment is NP-complete, except when the set of allowed relations satisfies one of the following six conditions:

1. Every relation holds when all variables are False.
2. Every relation holds when all variables are True.
3. Every relation is definable by a formula in conjunctive normal form in which each conjunct has at most one negated variable.
4. Every relation is definable by a formula in conjunctive normal form in which each conjunct has at most one unnegated variable.
5. Every relation is definable by a formula in conjunctive normal form in which each conjunct contains at most 2 literals.
6. Every relation is the set of solutions of a system of linear equations over the finite field GF(2).

In this paper, we show how this result can be derived from the algebraic structure of Boolean relations. This result allows us to describe an efficient test which can be applied to any set of Boolean relations in order to determine whether or not they lie within one of these tractable classes. The existence of such a test was left as an open question in Schaefer's 1978 paper [15].

2. Definitions and examples

2.1. Relations and relational structures

We first define the basic terminology of relations and functions.

Definition 2.1. For any set A and any natural number n, A^n denotes the set of all n-tuples of elements of A. Elements of A^n will be written $\langle a_1, a_2, \ldots, a_n \rangle$.

A subset of A^n is called an *n-ary relation over A*.[1]

For any binary relation R, the set $\{a \mid \exists \langle a, b \rangle \in R\}$ is called the *domain* of R, and the set $\{b \mid \exists \langle a, b \rangle \in R\}$ is called the *range* of R.

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

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

- Equality: $\square_A = \{\langle a, a' \rangle \in A^2 \mid a = a'\}$

[1] Technically, we should distinguish empty relations of different arities, but we shall neglect this special case, in order to simplify the presentation.

- Disequality: $\Diamond_A = \{\langle a, a' \rangle \in A^2 \mid a \neq a'\}$.

Definition 2.3. A *function* f from a set A to a set B, denoted $f : A \to B$, is a subset of $A \times B$ such that for each $a \in A$ there is exactly one $b \in B$ such that $\langle a, b \rangle \in f$. If $\langle a, b \rangle \in f$ then we write $f(a) = b$.

If it is the case that $f(a) = f(a')$ implies $a = a'$, then f is said to be *injective*.

Definition 2.4. An *n-ary operation* on a set A is a function $\varphi : A^n \to A$.

If $\langle \langle a_1, a_2, \ldots, a_n \rangle, a_{n+1} \rangle \in \varphi$ then we write $\varphi(a_1, a_2, \ldots, a_n) = a_{n+1}$.

If $\varphi(a, a, \ldots, a) = a$, for all $a \in A$, then φ is said to be *idempotent*.

If $\varphi(a_1, a_2, \ldots, a_n) \in \{a_1, a_2, \ldots, a_n\}$, for all $a_1, a_2, \ldots, a_n \in A$, then φ is said to be *conservative*.

We will sometimes want to refer to operations with special properties, as specified in the following definition.

Definition 2.5. Let φ be an n-ary operation from A^n to A.

- If $n = 1$ and φ is injective, then φ is called a *permutation*.
- If there exists an index $i \in \{1, 2, \ldots, n\}$ such that for all $\langle a_1, a_2, \ldots, a_n \rangle \in A^n$ we have $\varphi(a_1, a_2, \ldots, a_n) = f(a_i)$, where f is a non-constant unary operation on A, then φ is called *essentially unary*.

  If this f is the identity operation, then φ is called a *projection*.
- If $n = 2$ and for all $a_1, a_2, a_3 \in A$ we have $\varphi(\varphi(a_1, a_2), a_3) = \varphi(a_1, \varphi(a_2, a_3))$ (Associativity), and $\varphi(a_1, a_2) = \varphi(a_2, a_1)$ (Commutativity), then φ is called an AC operation.
- If $n \geqslant 3$ and there exists an index $i \in \{1, 2, \ldots, n\}$ such that for all $a_1, a_2, \ldots, a_n \in A$ such that $|\{a_1, a_2, \ldots, a_n\}| < n$ we have $\varphi(a_1, a_2, \ldots, a_n) = a_i$, but φ is not a projection, then φ is called a *semiprojection* [14, 16].
- If $n = 3$ and for all $a, a' \in A$ we have $\varphi(a, a, a') = \varphi(a, a', a) = \varphi(a', a, a) = a$, then φ is called a *majority operation*.

  The majority operation on A given by

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

  is called the *dual discriminator* on A [16], and will be denoted μ_A.
- If $n = 3$ and for all $a_1, a_2, a_3 \in A$ we have $\varphi(a_1, a_2, a_3) = a_1 + a_2 + a_3$, where $+$ is a binary operation on A such that $\langle A, + \rangle$ is an elementary Abelian 2-group [11], then φ is called a *generalised parity operation*.

In order to describe combinatorial problems in algebraic terms, we will make extensive use of the notion of a 'relational structure' [2, 11].

Definition 2.6. A *relational structure* is a pair, $\langle V, E_i (i \in I) \rangle$, consisting of a non-empty set, V, and a system, E_i, of finitary relations over V, indexed by the elements of I.

The set V is called the *universe* of the relational structure.

A relational structure $S = \langle V, E_i(i \in I)\rangle$ is called *finite* if V and I are finite sets. In this case we will sometimes write S as $\langle V, E_1, E_2, \ldots, E_{|I|}\rangle$.

Example 2.7. A (directed) graph is a relational structure with a single binary relation specifying which vertices are adjacent. It is usually written $\langle V, E\rangle$.

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

A graph in which the edges are labelled with elements of some set L can be written as a relational structure $\langle V, E_i(i \in L)\rangle$, where the relation E_i contains all edges labelled with i.

Definition 2.8. The *rank function* of a relational structure $\langle V, E_i(i \in I)\rangle$, is a function ρ from I to the set of non-negative integers, such that for all $i \in I$, $\rho(i)$ is the arity of E_i.

A relational structure S is *similar* to a relational structure T if they have the same rank function.

Definition 2.9. Let $S = \langle V, E_i(i \in I)\rangle$ and $S' = \langle V', E'_i(i \in I)\rangle$ be two similar relational structures, and let ρ be their common rank function.

A *homomorphism* from S to S' is a function $h : V \to V'$ such that, for all $i \in I$,

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

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

2.2. Combinatorial problems

In this section we will demonstrate that a wide variety of standard combinatorial problems can be conveniently expressed as subproblems of the following very general problem. This allows us to develop a common algebraic theory, in the remainder of the paper, which is applicable to all of these problems.

Definition 2.10. The *general combinatorial problem* (GCP) is the decision problem with

Instance: A pair of similar finite relational structures, $\langle S_1, S_2\rangle$.

Question: Is there a homomorphism from S_1 to S_2?

For any GCP instance $P = \langle S_1, S_2\rangle$, a homomorphism from S_1 to S_2 will be called a *solution* to P.

Example 2.11 (*GRAPH COLORABILITY*). An instance of the GRAPH COLORABILITY problem [4, 13] 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 2.7.

Example 2.12 (*CLIQUE*). An instance of the CLIQUE problem [4, 13] 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 $\langle K_q, G \rangle$.

Example 2.13 (*VERTEX COVER*). An instance of the VERTEX COVER problem [4, 13] consists of a graph $G = \langle V, E \rangle$ and an integer k. The question is whether there is a subset $V' \subseteq V$ with $|V'| \leqslant k$, such that for all $\langle v, w \rangle \in E$ either $v \in V'$ or $w \in V'$.

This can be expressed as the GCP instance $\langle K_{(|V|-k)}, \overline{G} \rangle$, where $\overline{G}$ is the complement graph $\langle V, \diamond_V - E \rangle$. (This formulation uses the fact that V' is a vertex cover if and only if $V - V'$ is an independent set, which corresponds to a clique in the complement graph [4, 13].)

Example 2.14 (*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 $\langle \langle V, \diamond_V \rangle, \langle M, \tilde{\diamond}_M \rangle \rangle$, where $\tilde{\diamond}_M = \{ \langle \langle v_1, v_2, \ldots, v_k \rangle, \langle v'_1, v'_2, \ldots, v'_k \rangle \rangle \in M^2 \mid v_i \neq v'_i, \ i = 1, 2, \ldots, k \}$.

Example 2.15 (*HAMILTONIAN CIRCUIT*). An instance of the HAMILTONIAN CIRCUIT problem [4] consists of a graph $G = \langle V, E \rangle$. The question is whether there is a 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 $\langle \langle V, C_V, \diamond_V \rangle, \langle V, E, \diamond_V \rangle \rangle$, where C_V is an arbitrary cyclic permutation on V and $\diamond_V$ is the disequality relation over V defined above. (The presence of the relation $\diamond_V$ in both relational structures simply ensures that any solution must be injective.)

Example 2.16 (*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 $\langle \langle V, E, \diamond_V \rangle, \langle V, B_k, \diamond_V \rangle \rangle$, where $B_k = \{ \langle v_i, v_j \rangle \in V^2 \mid |i - j| \leqslant k \}$ and $\diamond_V$ is the disequality relation over V defined above.

Example 2.17 (*GRAPH ISOMORPHISM*). An instance of the GRAPH ISOMORPHISM problem [4] 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 $\langle\langle V,E,\overline{E}\rangle,\langle V',E',\overline{E'}\rangle\rangle$, where $\overline{E}=\Diamond_V-E$ and $\overline{E'}=\Diamond_{V'}-E'$.

Example 2.18 (*UNDIRECTED GRAPH REACHABILITY*). An instance of the UNDIRECTED GRAPH REACHABILITY problem [13] consists of an undirected graph $G=\langle V,E\rangle$ and a pair of vertices, $v,w\in V$. The question is whether there is a path in G which connects v to w.

The complementary problem (whether there is *no* path connecting v and w), can be expressed as the GCP instance $\langle\langle V,E,\{\langle v\rangle\},\{\langle w\rangle\}\rangle,\langle\{0,1\},\Box_{\{0,1\}},\{\langle 0\rangle\},\{\langle 1\rangle\}\rangle\rangle$.

Example 2.19 (*SATISFIABILITY*). An instance of the SATISFIABILITY problem [4] consists of a formula, $\mathscr{F}$, in propositional logic, which is the conjunction of a set of clauses, C. Each clause in C is a disjunction of literals, where a literal is either a propositional variable or its negation. The question is whether there is an assignment of truth values to the variables in $\mathscr{F}$ such that $\mathscr{F}$ is true.

This can be expressed as the GCP instance $\langle\langle V,E_c(c\in C)\rangle,\langle\{0,1\},R_c(c\in C)\rangle\rangle$, where V is the set of propositional variables used in $\mathscr{F}$, each $E_c=\{\langle x_1,x_2,\ldots,x_{\rho(c)}\rangle\}$, where $x_1,x_2,\ldots,x_{\rho(c)}$ are the variables appearing in the clause c, and

$$R_c=\{\langle h(x_1),h(x_2),\ldots,h(x_{\rho(c)})\rangle \mid h \text{ is an assignment of truth values satisfying } c\}.$$

Example 2.20 (*CONSTRAINT SATISFACTION*). An instance of the CONSTRAINT SATISFACTION problem [10, 12] consists of a set of variables V, a domain of values D, and a list of constraints $C(S_1),C(S_2),\ldots,C(S_m)$, where each S_i is an ordered subset of V and the constraint $C(S_i)$ is a set of tuples specifying the allowed combinations of values for variables in S_i. The question is whether there is an assignment of values from D to the variables in V such that every constraint is satisfied.

This can be expressed as the GCP instance

$$\langle\langle V,S_1,S_2,\ldots,S_m\rangle,\langle D,C(S_1),C(S_2),\ldots,C(S_m)\rangle\rangle.$$

2.3. Subproblems with restricted constraints

It is clear from the examples above that GCP is NP-hard.

In the rest of the paper we shall examine how restricting the relational structures allowed in problem instances affects the complexity of this decision problem.

In particular, we shall investigate the effect of restricting the relational structures allowed in the *second* component of each problem instance. We therefore define the following family of subproblems of GCP.

Definition 2.21. Let Γ be a set of relations. **GCP**(Γ) is the decision problem with

Instance: A pair of similar relational structures, $\langle S_1,S_2\rangle$, in which the relations of S_2 are elements of Γ.

Question: Is there a homomorphism from S_1 to S_2?

Example 2.22. Consider the relation C of arity 4 defined as follows:

$$C = \{\ (0,1,1,1),\ (1,0,0,1),\ (1,0,1,0),\ (1,0,1,1),\ (0,1,1,0)\ \}.$$

GCP($\{C\}$) is the subproblem of GCP containing all instances $\langle S_1, S_2\rangle$, such that the only relation occurring in S_2 is C.

Example 2.23 (*GRAPH k-COLORABILITY*). Let $\Diamond_V$ be the disequality relation over a set V with $|V| = k$.

Examples 2.7 and 2.11 show that GCP($\{\Diamond_V\}$) corresponds to the standard GRAPH k-COLORABILITY problem [4].

In order to define subproblems of the SATISFIABILITY problem, we need to consider relations over $\{0,1\}$ which correspond to sets of models of certain Boolean expressions. We therefore make the following definition.

Definition 2.24. For any positive integer k, the sets of relations Δ_k and Δ_k^H are defined as follows:

$$\Delta_k = \{\{0,1\}^k - \{t\} \mid t \in \{0,1\}^k\},$$
$$\Delta_k^H = \{\{0,1\}^k - \{t\} \mid t \in \{0,1\}^k,\ t \text{ has at most one 0 entry}\}.$$

Example 2.25. The elements of Δ_k are precisely the relations which correspond to models of disjunctive clauses of length k involving k distinct variables.

For example, Δ_2 is the set which contains the following 4 relations:

- $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\}$ (models of $\neg x_1 \vee x_2$);
- $\{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle\}$ (models of $x_1 \vee \neg x_2$);
- $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\}$ (models of $\neg x_1 \vee \neg x_2$);
- $\{\langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle\}$ (models of $x_1 \vee x_2$).

The elements of Δ_k^H are precisely the relations which correspond to models of disjunctive *Horn* clauses of length k involving k distinct variables.

For example, Δ_2^H is the set containing the first three relations of Δ_2 listed above.

Example 2.26 (*k-SATISFIABILITY*). Examples 2.19 and 2.25 show that GCP(Δ_k) corresponds to the k-SATISFIABILITY problem [4].

Example 2.27 (*HORN-CLAUSE-SATISFIABILITY*). Examples 2.19 and 2.25 show that GCP($\bigcup_{k=1}^{\infty} \Delta_k^H$) corresponds to the HORN-CLAUSE-SATISFIABILITY problem [13].

Example 2.28 (*NOT-ALL-EQUAL SATISFIABILITY*). Let N be the ternary Boolean relation containing the following tuples:

$$N = \{\langle 0,0,1\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle, \langle 1,0,1\rangle, \langle 0,1,1\rangle\}.$$

Example 2.19 shows that GCP($\{N\}$) corresponds to the NOT-ALL-EQUAL SATISFIABILITY problem [4, 15].

Example 2.29 (*ONE-IN-THREE SATISFIABILITY*). Let T be the ternary Boolean relation containing the following tuples:

$$T = \{\langle 0,0,1\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle\}.$$

Example 2.19 shows that GCP($\{T\}$) corresponds to the ONE-IN-THREE SATISFIABILITY problem [4, 15].

3. Reductions between problems

In this section we investigate reductions between the subproblems of GCP defined above.

In order to quantify the complexity of these reductions we need to define the representation and 'size' of a problem instance. In order to do this, we shall assume, for simplicity, that the (finite) relations in a problem instance are specified by giving an explicit list of all their tuples. The size of a problem instance will then be taken to be the length of a string encoding the pair of relational structures using some standard encoding.

Proposition 3.1. *Let Γ be a set of relations over a set D.*
GCP($\Gamma \cup \{\square_D\}$) *is polynomial-time reducible to* GCP(Γ).

Proof. Let $P = \langle S_1, S_2\rangle$ be any instance of GCP($\Gamma \cup \{\square_D\}$). We will modify P to construct an instance P' of GCP(Γ), as follows.

For each relation E_i of S_1, and corresponding relation C_i of S_2, if $C_i = \square_D$ then

- for each $\langle v_1, v_2\rangle \in E_i$, remove $\langle v_1, v_2\rangle$ from E_i, and replace all remaining occurences of v_1 in all the relations of S_1 with v_2 (including any remaining occurences in E_i);
- remove E_i from S_1 and C_i from S_2.

Clearly, this construction can be carried out in polynomial-time in the size of P, and P' has a solution if and only if P has a solution. Hence, we have established a polynomial time reduction from GCP($\Gamma \cup \{\square_D\}$) to GCP($\Gamma$). □

It is currently an open question whether GCP($\Gamma \cup \{\square_D\}$) is always reducible to GCP(Γ) in logarithmic space. If this were the case, then Example 2.18 indicates that UNDIRECTED GRAPH REACHABILITY would be solvable in deterministic logarithmic-space, which is thought to be unlikely [13].

We now establish a much more powerful result, which gives very general conditions under which logarithmic-space reductions can be obtained. In order to describe these conditions, we first define the set of relations which can be 'generated' from a given set of relations.

Definition 3.2. The set of relations which is *generated* by a set of relations, Γ over a set D, denoted Γ^*, is defined to be the smallest set of relations such that:

1. $\Gamma \subseteq \Gamma^*$;
2. (**permutation**) For any $C \in \Gamma^*$ of arity r, and any permutation σ with domain $\{1,2,\ldots,r\}$, $\{\langle x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(r)}\rangle \mid \langle x_1, x_2, \ldots, x_r\rangle \in C\} \in \Gamma^*$;
3. (**extension**) For any $C \in \Gamma^*$ of arity r, $\{\langle x_1, x_2, \ldots, x_r, x_{r+1}\rangle \mid \langle x_1, x_2, \ldots, x_r\rangle \in C,\ x_{r+1} \in D\} \in \Gamma^*$;
4. (**truncation**) For any $C \in \Gamma^*$ of arity r (>1), $\{\langle x_1, x_2, \ldots, x_{r-1}\rangle \mid \exists \langle x_1, x_2, \ldots, x_r\rangle \in C\} \in \Gamma^*$;
5. (**intersection**) For all $C_1, C_2 \in \Gamma^*$, $C_1 \cap C_2 \in \Gamma^*$.

Example 3.3. Reconsider the set Δ_2, defined in Definition 2.24, consisting of all binary Boolean relations which can be defined by a binary disjunction involving 2 distinct variables.

- Δ_2^* contains the following unary relations:
  - $\{\langle 0\rangle, \langle 1\rangle\}$ (truncation of $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\} \in \Delta_2$);
  - $\{\langle 0\rangle\}$ (truncation of $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\} \cap \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\} \in \Delta_2^*$);
  - $\{\langle 1\rangle\}$ (truncation of $\{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle\} \cap \{\langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle\} \in \Delta_2^*$).
- Δ_2^* contains all possible binary relations over $\{0,1\}$ (which can all be obtained by intersection from elements of Δ_2, or by extension from the first unary relation above.)
- Δ_2^* contains a large number of ternary relations, including the following:
  - $\{\langle 0,0,0\rangle, \langle 1,1,0\rangle, \langle 0,0,1\rangle, \langle 1,1,1\rangle\}$ (extension of $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\} \cap \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle\} \in \Delta_2^*$);
  - $\{\langle 0,0,0\rangle, \langle 1,0,1\rangle, \langle 0,1,0\rangle, \langle 1,1,1\rangle\}$ (permutation of the above relation);
  - $\{\langle 0,0,0\rangle, \langle 1,1,1\rangle\}$ (intersection of the above 2 relations).

  It is left as an exercise for the reader to establish which other ternary relations belong to Δ_2^* (but see Example 4.9 below).

Now we show how the fact that one set of relations can be generated from another can be used to obtain a logarithmic-space reduction between problems.

Theorem 3.4. *Let Γ and Γ_0 be sets of relations over a set D. If $\Gamma_0 \subseteq \Gamma^*$, and Γ_0 is finite, then* GCP(Γ_0) *is logarithmic-space reducible to* GCP(Γ).

Proof. Assume that $\Gamma_0 \subseteq \Gamma^*$, and Γ_0 is finite. By Definition 3.2, this implies that Γ_0 can be obtained from Γ by a finite sequence of permutations, extensions, truncations and intersections. Let Σ be a minimal sequence of these operations which is sufficient

to construct all the elements of Γ_0 from Γ. We shall prove that GCP(Γ_0) is logarithmic-space reducible to GCP(Γ) by induction on the length of Σ.

Let $P = \langle S_1, S_2 \rangle$ be any instance of GCP(Γ_0).

If Σ is empty, then every relation in S_2 is an element of Γ. Hence the result holds when Σ is empty.

Now, assume that Σ contains $n > 0$ operations, and assume that the result holds for all shorter sequences.

Let Σ' be the sequence consisting of the first $n-1$ operations of Σ, and let Γ_0' be the set of relations constructed from Γ by the operations in Σ'.

We may assume, without loss of generality, that every $C \in \Gamma_0$ occurs in S_2, since for each C which is not a relation of S_2 we can simply extend S_2 by adding the relation C and extend S_1 by adding a corresponding empty relation. Since Γ_0 is finite, this process can be carried out in constant space.

There are 4 cases to consider, depending on the type of the final operation of Σ.

- (**permutation**) If the final operation of Σ is a permutation, then S_2 has a relation C which is obtained by permuting some element of Γ_0'. By applying the inverse permutation to C and to the corresponding relation in S_1 we obtain a new problem instance P' with the same set of solutions as P but belonging to GCP(Γ_0').
- (**extension**) If the final operation of Σ is an extension, then S_2 has a relation C which is obtained by extending some element of Γ_0'. By truncating C and the corresponding relation in S_1 we obtain a new problem instance P' with the same set of solutions as P but belonging to GCP(Γ_0').
- (**truncation**) If the final operation of Σ is a truncation, then S_2 has a relation C which is obtained by truncating some element $C' \in \Gamma_0'$. We construct a new problem instance P' by modifying P as follows. Replace C with C', and replace the corresponding relation E in S_1 with a new relation E' which is constructed as follows. For each $e \in E$, add a new element v_e to the universe of S_1, and set

  $$E' = \{\langle v_1, v_2, \ldots, v_r, v_e \rangle \mid e = \langle v_1, v_2, \ldots, v_r \rangle \in E\}.$$

  It follows from the above construction that P' has a solution if and only if P has a solution, but P' belongs to GCP(Γ_0').
- (**intersection**) If the final operation of Σ is an intersection, then S_2 has a relation C which is obtained by intersecting two elements C_1 and C_2 of Γ_0'. By replacing C with the pair of relations C_1, C_2, and replacing the corresponding relation E of S_1 with two copies of E, we obtain a new problem instance P' with the same set of solutions as P but belonging to GCP(Γ_0').

In all cases, we have shown that there is a logarithmic-space reduction from P to a problem instance P' in GCP(Γ_0'). Hence, by the inductive hypothesis, and the transitivity of logarithmic-space reductions [13], there is a logarithmic-space reduction from P to some problem instance in GCP(Γ). □

In order to use this result we need to be able to establish whether a given finite set of relations, Γ_0, can be generated from another set of relations Γ. In the next section

we shall show that this can be determined from simple algebraic properties of Γ and Γ_0.

4. Algebraic properties of relations

The algebraic properties described in this section concern closure operations on sets of relations.

First, we note that any operation on a set can be used to define an operation on tuples over that set, by applying the operation at each coordinate position separately, as described in the following definition:

Definition 4.1. Let R be a relation of arity r over a set D, and let φ be a k-ary operation on D.

For all $t_1, t_2, \ldots, t_k \in R$, (not necessarily all distinct) where $t_i = \langle d_{i1}, d_{i2}, \ldots, d_{ir} \rangle$, we define the tuple $\varphi(t_1, t_2, \ldots, t_k)$ as follows:

$$\varphi(t_1, t_2, \ldots, t_k) = \langle \varphi(d_{11}, d_{21}, \ldots, d_{k1}), \varphi(d_{12}, d_{22}, \ldots, d_{k2}), \ldots, \varphi(d_{1r}, d_{2r}, \ldots, d_{kr}) \rangle.$$

We define the relation $\varphi(R)$ to be the set $\{\varphi(t_1, t_2, \ldots, t_k) \mid t_1, t_2, \ldots, t_k \in R\}$.

Finally, we say that R *is closed under* φ if $\varphi(R) \subseteq R$.

Example 4.2. The (unique) majority operation on the set $\{0,1\}$ is the dual discriminator, $\mu_{\{0,1\}}$ (see Definition 2.5). We will denote this operation simply by μ.

The relation C given in Example 2.22 is closed under μ, since applying the μ operation to any 3 elements of C, as described in Definition 4.1, yields an element of C. For example,

$$\begin{aligned}\mu(\langle 0,1,1,1\rangle, \langle 1,0,0,1\rangle, \langle 1,0,1,0\rangle) &= \langle \mu(0,1,1), \mu(1,0,0), \mu(1,0,1), \mu(1,1,0)\rangle \\ &= \langle 1,0,1,1\rangle \in C.\end{aligned}$$

Example 4.3. Let $\diamond_D$ be the disequality relation over a set D, as defined in Definition 2.2, and let φ be any injective unary operation on D (i.e. a permutation).

For all $d, d' \in D$, we have

$$(\langle d, d' \rangle \in \diamond_D) \Leftrightarrow (d \neq d') \Leftrightarrow (\varphi(d) \neq \varphi(d')) \Leftrightarrow \varphi(\langle d, d' \rangle) \in \diamond_D.$$

Hence $\diamond_D$ is closed under all permutations on D.

By a similar argument, the relation N defined in Example 2.28 is closed under all permutations on $\{0,1\}$.

If a set of relations is closed under some unary operation, then we can use this fact to obtain a logarithmic-space equivalence between problems, as follows:

Proposition 4.4. *Let Γ be a finite set of relations over a set D, let φ be a unary operation on D, and let $\varphi(\Gamma)$ be defined as follows:*

$$\varphi(\Gamma) = \{\varphi(C) \mid C \in \Gamma\}.$$

If every $C \in \Gamma$ is closed under φ, then GCP(Γ) *is logarithmic-space equivalent to* GCP($\varphi(\Gamma)$).

Proof. Any instance $P = \langle S_1, S_2 \rangle$ of GCP(Γ) can be transformed in logarithmic-space to an instance P' of GCP($\varphi(\Gamma)$), by replacing each relation C of S_2 by the relation $\varphi(C)$. If C is closed under φ, then $\varphi(C) \subseteq C$, so any solution to P' is a solution to P. Conversely, if h is a solution to P, then φh, (the composition of h and φ), is a solution to P'. Hence, the transformation which maps P to P' is a logarithmic-space reduction.

Similarly, there is a logarithmic-space reduction from any instance P' of GCP($\varphi(\Gamma)$) to an instance P of GCP(Γ). □

Definition 4.5. Let Γ be a set of relations over a set D. Define $\Gamma^{\triangleright}$ to be the set of all operations, φ, on D such that every relation in Γ is closed under φ.

Definition 4.6. Let Φ be a set of operations on a set D.

Define $\Phi^{\triangleleft}$ to be the set of all relations over D which are closed under every element of Φ.

The mappings $^{\triangleright}$ and $^{\triangleleft}$ establish a Galois connection between sets of relations and sets of operations [2, 11]. By making use of this Galois connection we can obtain considerable insight into the relationship between different combinatorial problems, as the next results indicate. First, we show that if all the relations in some set are closed under some operation, then so are all the relations generated by that set.

Lemma 4.7. *Let Γ be a set of relations. If $\varphi \in \Gamma^{\triangleright}$ then $\varphi \in \Gamma^{*\triangleright}$.*

Proof. Follows from Definition 3.2, since the property of being closed under φ is preserved by permutation, extension, truncation and intersection. □

Example 4.8. It was shown in Example 4.3 that $\Diamond_D$ is closed under all permutations on D. Hence, by Lemma 4.7, every relation in $\{\Diamond_D\}^*$ is closed under all permutations on D. The only unary relation with this property is $D^1 = \{\langle d \rangle \mid d \in D\}$, and the only binary relations with this property are D^2, $\Diamond_D$, and $\Box_D$. Hence, $\{\Diamond_D\}^*$ contains no other unary or binary relations.

Example 4.9. Reconsider the set Δ_2, defined in Definition 2.24. Some elements of Δ_2^* were described in Example 3.3. Using Lemma 4.7, we can obtain more information about Δ_2^*.

It is easily verified that every relation in Δ_2 is closed under the unique majority operation μ on $\{0, 1\}$. Hence, by Lemma 4.7, every relation in Δ_2^* is closed under μ.

A routine calculation shows that 166 of the 256 possible ternary relations over $\{0,1\}$ are closed under μ. Hence Δ_2^* contains at most these 166 ternary relations over $\{0,1\}$.

We now establish the central result of this section, which shows that the set of relations generated by a given set of relations is completely determined by the set of operations under which it is closed. (This result was suggested by Theorem 2 of [5]).

Theorem 4.10. *For any set of relations, Γ, over a finite set D, such that $\square_D \in \Gamma^*$,*

$$\Gamma^* = \Gamma^{\triangleright\triangleleft}.$$

Proof. We first show that $\Gamma^* \subseteq \Gamma^{\triangleright\triangleleft}$. By Lemma 4.7 we have $\Gamma^{\triangleright} \subseteq \Gamma^{*\triangleright}$. Hence, $\Gamma^{\triangleright\triangleleft} \supseteq \Gamma^{*\triangleright\triangleleft} \supseteq \Gamma^*$.

Now we show, conversely, that $\Gamma^{\triangleright\triangleleft} \subseteq \Gamma^*$. Let R be any element of $\Gamma^{\triangleright\triangleleft}$, let $m = |R|$ and let r be the arity of R.

Let M be an m by $|D|^m$ matrix over D in which the columns are all possible m-tuples over D, and construct a new relation S, from R and M, by concatenating each element of R to a distinct row of M. The arity of S is therefore $r + |D|^m$, which we denote by s.

Now define $\hat{S} = \bigcap \{C \in \Gamma^* \mid S \subseteq C\}$. Note that $\hat{S}$ is a finite intersection of elements of Γ^*, hence $\hat{S} \in \Gamma^*$, by Definition 3.2.

Denote the elements of S by $t_1, t_2, \ldots, t_m$, where $t_i = \langle t_{i1}, t_{i2}, \ldots, t_{is}\rangle$. Let $t_0 = \langle t_{01}, t_{02}, \ldots, t_{0s}\rangle$ be any element of $\hat{S}$, and consider the binary relation ϕ defined by

$$\phi = \{\langle\langle t_{11}, t_{21}, \ldots, t_{m1}\rangle, t_{01}\rangle, \langle\langle t_{12}, t_{22}, \ldots, t_{m2}\rangle, t_{02}\rangle, \ldots, \langle\langle t_{1s}, t_{2s}, \ldots, t_{ms}\rangle, t_{0s}\rangle\}.$$

The domain of ϕ is D^m (by the construction of S).

We claim that ϕ is a function from D^m to D. To establish this claim we need to show that for any two values of j and k in the range $1,2,\ldots,s$, if $\langle t_{1j}, t_{2j}, \ldots, t_{mj}\rangle = \langle t_{1k}, t_{2k}, \ldots, t_{mk}\rangle$ then $t_{0j} = t_{0k}$. But this follows from the fact that if every tuple in S is invariant under the permutation which exchanges position j with position k, then every element of $\hat{S}$ is invariant under the same permutation, because $\square_D \in \Gamma^*$ (by assumption), so Γ^* contains all relations of the form $E_{jk} = \{\langle d_1, d_2, \ldots, d_s\rangle \in D^s \mid d_j = d_k\}$.

We now show that $\phi \in \Gamma^{\triangleright}$. Assume for contradiction that some $C \in \Gamma$ is not closed under ϕ. By appropriate extensions, truncations and permutations of C we could then obtain a relation $C' \in \Gamma^*$ which was a superset of S, but remained not closed under ϕ. However, this implies that $t_0 \notin C'$, hence $t_0 \notin \hat{S}$, which contradicts the choice of t_0.

Now, define $\hat{R}$ to be the truncation of $\hat{S}$ to the first r coordinates. Note that $\hat{R} \in \Gamma^*$, by Definition 3.2.

Since $\phi \in \Gamma^{\triangleright}$, we know that R is closed under ϕ, by the choice of R, hence the truncation of t_0 to the first r coordinates is an element of R. But t_0 was arbitrary, so $R = \hat{R} \in \Gamma^*$, and the result follows. □

Corollary 4.11. *Let Γ and Γ_0 be sets of relations over a finite set D, such that Γ_0 is finite. If $\Gamma^{\triangleright} \subseteq \Gamma_0^{\triangleright}$, then* GCP($\Gamma_0$) *is polynomial-time reducible to* GCP(Γ).

If, in addition, $\square_D \in \Gamma^*$, then GCP(Γ_0) *is logarithmic-space reducible to* GCP(Γ).

Proof. If $\Gamma^{\triangleright} \subseteq \Gamma_0^{\triangleright}$ then, by Theorem 4.10, we have $(\Gamma_0 \cup \{\square_D\})^* \subseteq (\Gamma \cup \{\square_D\})^*$, and hence $\Gamma_0 \subseteq (\Gamma \cup \{\square_D\})^*$.

Hence, by Theorem 3.4, GCP(Γ_0) is logarithmic-space reducible to GCP($\Gamma \cup \{\square_D\}$), and by Proposition 3.1, GCP(Γ_0) is polynomial-time reducible to GCP(Γ).

Furthermore, if $\square_D \in \Gamma^*$, then a further application of Theorem 3.4 shows that GCP(Γ_0) is logarithmic-space reducible to GCP(Γ). □

This corollary demonstrates that the complexity of GCP(Γ) is effectively determined by $\Gamma^{\triangleright}$. The next theorem uses a general result from universal algebra [14, 16] to show that the possible choices for $\Gamma^{\triangleright}$ are quite limited.

Theorem 4.12. *For any set of relations Γ, on a finite set, at least one of the following conditions must hold:*

1. *$\Gamma^{\triangleright}$ contains a constant operation;*
2. *$\Gamma^{\triangleright}$ contains an idempotent binary operation (which is not a projection);*
3. *$\Gamma^{\triangleright}$ contains a majority operation;*
4. *$\Gamma^{\triangleright}$ contains a generalised parity operation;*
5. *$\Gamma^{\triangleright}$ contains a semiprojection;*
6. *$\Gamma^{\triangleright}$ contains essentially unary operations only.*

Proof. The set of operations $\Gamma^{\triangleright}$ contains all projections and is closed under composition, hence it constitutes a 'clone' [2, 11, 16]. It was shown in [14] that any clone on a finite set must contain a minimal clone, and that any minimal clone contains either

1. a constant operation; or
2. an idempotent binary operation (which is not a projection); or
3. a majority operation; or
4. a generalised parity operation; or
5. a semiprojection; or
6. a non-identity unary operation.

Furthermore, if $\Gamma^{\triangleright}$ contains any operations which are not essentially unary, then it is straightforward to show, by considering such an operation of the smallest possible arity, that $\Gamma^{\triangleright}$ contains an operation in one of the first five of these classes [16]. □

In the next section we show that each of these possibilities for $\Gamma^{\triangleright}$ is associated in a very natural way with a well-known complexity class.

5. Determining complexity

We now examine each of the 6 possibilities for $\Gamma^{\triangleright}$ described in Theorem 4.12, and investigate the implications for the complexity of the corresponding problem class,

GCP(Γ). Throughout this section, we shall assume that Γ is a finite set of relations over a finite set D with $|D| \geqslant 2$.

Proposition 5.1. *If $\Gamma^{\triangleright}$ contains a constant operation, then* GCP(Γ) *can be solved in constant space.*

Proof. If $\Gamma^{\triangleright}$ contains the constant operation which returns the value d, then every non-empty element of Γ must contain the tuple $\langle d, d, \ldots, d\rangle$. Hence, any problem instance $P = \langle S_1, S_2\rangle$ in GCP(Γ) either contains an empty relation in S_2 (associated with a non-empty relation in S_1), in which case it has no solution, or else has a solution (the constant function with value d). The decision problem can therefore be solved in constant space. □

For binary operations we shall restrict our attention to the case of idempotent AC operations (see Definition 2.5).

Proposition 5.2. *If $\Gamma^{\triangleright}$ contains an idempotent AC binary operation , φ, then* GCP(Γ) *is in* **P**.

If, in addition, $\Gamma^{\triangleright} = \{\varphi\}^{\triangleleft\triangleright}$ and $\square_D \in \Gamma^*$, then GCP(Γ) *is* **P**-*complete.*

Proof. Theorem 16 of [6] states that the CONSTRAINT SATISFACTION problem is solvable in polynomial time if all the constraints are closed under some idempotent AC operation. Using Example 2.20, this implies that GCP(Γ) is solvable in polynomial time, when $\Gamma^{\triangleright}$ contains an idempotent AC binary operation.

Furthermore, we may assume without loss of generality, that $\{0,1\} \subseteq D$, and $\varphi(v_1, v_2) = v_1 \wedge v_2$, for all $v_1, v_2 \in \{0,1\}$.

Now, set $\Gamma_H = \Delta_1^H \cup \Delta_2^H \cup \Delta_3^H$. It is easy to show that this set of relations is closed under conjunction, so if we consider Γ_H as a set of relations over D, then $\varphi \in \Gamma_H^{\triangleright}$. This implies that $\{\varphi\}^{\triangleleft\triangleright} \subseteq \Gamma_H^{\triangleright\triangleleft\triangleright}$, which is equal to $\Gamma_H^{\triangleright}$ by the general properties of Galois connections [2].

Hence if $\Gamma^{\triangleright} = \{\varphi\}^{\triangleleft\triangleright}$, then $\Gamma^{\triangleright} \subseteq \Gamma_H^{\triangleright}$, so by Corollary 4.11, if $\square_D \in \Gamma^*$, then GCP(Γ_H) is logarithmic-space reducible to GCP(Γ). However, GCP(Γ_H) corresponds to the HORN-CLAUSE-SATISFIABILITY problem with at most 3 variables per clause, which is **P**-complete [8]. Hence, GCP(Γ) is **P**-complete. □

For majority operations we restrict our attention to the dual discriminator operation, μ_D (see Definition 2.5).

Proposition 5.3. *If $\Gamma^{\triangleright}$ contains the dual discriminator operation, μ_D, then* GCP(Γ) *is in* **NL**.

If, in addition, $\Gamma^{\triangleright} = \{\mu_D\}^{\triangleleft\triangleright}$ and $\square_D \in \Gamma^*$, then GCP(Γ) *is* **NL**-*complete.*

Proof. Let Γ be a set of relations which is closed under μ_D. Propositions 10 and 12 of [6] together imply that GCP(Γ) can be reduced to GCP(Γ'), where Γ' contains only

binary '0/1/all' relations over D, as defined in [3] (see also Proposition 5.3 of [16]). These relations are also defined in [9], where they are referred to as 'implicational' relations. They have the special property that each element of the domain is either related to a unique element of the range, or else is related to every element of the range, and *vice versa*.

A polynomial-time algorithm for GCP(Γ') is given in [9], which is essentially a generalisation of the standard algorithm for 2-SATISFIABILITY. Informally, the algorithm proceeds as follows: while there are unassigned variables, choose an unassigned variable, v, choose some value to be assigned to v, and then propagate the consequences of this decision to all other variables whose value can now be determined. If these consequences lead to a contradiction, then undo these assignments and choose a different value for v, or if all possible values have been tried then report that there are no solutions (see [9] for details).

This algorithm will only report that there are no solutions if there is some variable for which there are chains of implications starting at each possible value which lead to a contradiction. This can be represented as an instance of the DIRECTED GRAPH REACHABILITY problem, where the vertices of the graph correspond to (variable, value) pairs, and the edges of the graph represent implications. Since DIRECTED GRAPH REACHABILITY is in **NL** [13], and **NL** is closed under complementation, it follows that GCP(Γ) is in **NL**.

Furthermore, we may assume, without loss of generality, that $\{0,1\} \subseteq D$, and $\mu_D(v_1, v_2, v_3) = \mu_{\{0,1\}}(v_1, v_2, v_3)$ for all $v_1, v_2, v_3 \in \{0,1\}$.

Now, consider the set of relations Δ_2 defined in Definition 2.24. It is easy to show that $\mu_{\{0,1\}} \in \Delta_2^{\triangleright}$, so if we consider Δ_2 as a set of relations over D, then $\mu_D \in \Delta_2^{\triangleright}$. This implies that $\{\mu_D\}^{\triangleleft\triangleright} \subseteq {\Delta_2}^{\triangleright\triangleleft\triangleright}$, which is equal to $\Delta_2^{\triangleright}$, by the general properties of Galois connections [2].

Hence if $\Gamma^{\triangleright} = \{\mu_D\}^{\triangleleft\triangleright}$, then $\Gamma^{\triangleright} \subseteq \Delta_2^{\triangleright}$, so by Corollary 4.11, if $\square_D \in \Gamma^*$, then GCP(Δ_2) is logarithmic-space reducible to GCP(Γ). However, GCP(Δ_2) corresponds to the 2-SATISFIABILITY problem which is **NL**-complete [13]. Hence GCP(Γ) is **NL**-complete. □

Proposition 5.4. *If $\Gamma^{\triangleright}$ contains a generalised parity operation, φ, then* GCP(Γ) *is in* **P**.

Proof. If $\Gamma^{\triangleright}$ contains a generalised parity operation, then every element of Γ is an affine set over the finite field with 2^n elements, GF(2)^n, and hence contains precisely the solutions to some system of linear equations over GF(2).

Hence, any problem in GCP(Γ) can be solved in polynomial time, by a standard technique of linear algebra, such as Gaussian elimination. □

Proposition 5.5. *If $\Gamma^{\triangleright}$ contains projections and semiprojections only, then* GCP(Γ) *is* **NP**-*complete.*

Proof. Assume, without loss of generality, that $\{0,1\} \subseteq D$, and let Let Δ_3 be the set of ternary Boolean relations defined in Definition 2.24. It is easy to show that Δ_3 is closed under all projections and semiprojections on D.

Hence, if $\Gamma^{\triangleright}$ contains only projections and semiprojections, then by Corollary 4.11, GCP(Δ_3) must be polynomial-time reducible to GCP(Γ). However, GCP(Δ_3) corresponds to the 3-SATISFIABILITY problem (Example 2.26) which is **NP**-complete [4]. Hence GCP(Γ) is **NP**-complete. □

Proposition 5.6. *If $\Gamma^{\triangleright}$ contains essentially unary operations only, then* GCP(Γ) *is* **NP**-*complete.*

Proof. First we note that if $\Gamma^{\triangleright}$ contains an essentially unary operation φ, then it also contains the corresponding (non-constant) unary operation f.

If f is not injective, then $f(D) = \{f(d) \mid d \in D\}$ is strictly smaller than D. Now let f be any unary function in $\Gamma^{\triangleright}$ such that $|f(D)|$ is minimal and set $f(\Gamma) = \{f(C) \mid C \in \Gamma\}$. It follows that $f(\Gamma)^{\triangleright}$ contains only permutations.

Since $\Gamma^{\triangleright}$ contains no constant functions (by assumption), we know that $|f(D)| > 1$. There are 2 cases to consider.

- If $|f(D)| = 2$ then we may assume without loss of generality that $f(D) = \{0,1\}$. By Example 4.3, we know that $f(\Gamma)^{\triangleright} \subseteq \{N\}^{\triangleright}$ where N is the ternary relation defined in Example 2.28. Hence, by Example 2.28 and Corollary 4.11, the NOT-ALL-EQUAL SATISFIABILITY problem is polynomial-time reducible to GCP($f(\Gamma)$). Since the NOT-ALL-EQUAL SATISFIABILITY problem is **NP**-complete [15, 4], this implies that GCP($f(\Gamma)$) is **NP**-complete.
- If $|f(D)| > 2$ then, by Example 4.3, we know that $f(\Gamma)^{\triangleright} \subseteq \{\Diamond_{f(D)}\}^{\triangleright}$. Hence, by Example 2.23 and Corollary 4.11, the GRAPH $|f(D)|$-COLORABILITY problem is polynomial-time reducible to GCP($f(\Gamma)$). Since the GRAPH k-COLORABILITY problem is **NP**-complete for $k > 2$ [4], this implies that GCP($f(\Gamma)$) is **NP**-complete.

Finally, by Proposition 4.4, GCP($f(\Gamma)$) is logarithmic-space reducible to GCP(Γ), and hence GCP(Γ) is **NP**-complete. □

Now that we have shown that the complexity of GCP(Γ) is determined by $\Gamma^{\triangleright}$, it is natural to ask how to calculate $\Gamma^{\triangleright}$. The surprising answer to this question is that for each possible arity, the elements of $\Gamma^{\triangleright}$ of that arity are precisely the solutions to a particular problem instance in GCP(Γ), as we shall now show.

In order to state this result concisely, we will make use of the notion of the product of relational structures [11].

Definition 5.7. If $S = \langle V, E_i (i \in I) \rangle$ is a relational structure, then for any natural number n, the product structure, S^n is the relational structure $\langle V^n, E_i' (i \in I) \rangle$, where $\langle \langle v_{11}, v_{12}, \ldots, v_{1n} \rangle, \langle v_{21}, v_{22}, \ldots, v_{2n} \rangle, \ldots, \langle v_{\rho(i)1}, v_{\rho(i)2}, \ldots, v_{\rho(i)n} \rangle \rangle \in E_i'$ if and only if $\langle v_{1j}, v_{2j}, \ldots, v_{\rho(i)j} \rangle \in E_i$ for $j = 1, 2, \ldots, n$.

Proposition 5.8. *For any relational structure* $S = \langle V, E_i(i \in I)\rangle$, *and any operation* $\varphi : V^n \to V$,

$$\varphi \in (E_i(i \in I))^{\triangleright} \Leftrightarrow \varphi \in \mathrm{Hom}(S^n, S)$$

Proof. Follows immediately from Definitions 4.5 and 4.1, and Definition 5.7. □

This result implies that for any finite set of relations, Γ, over a set D, the operations of arity n under which Γ is closed are precisely the solutions to the GCP instance $\langle S^n, S\rangle$, where $S = \langle D, \Gamma\rangle$.

6. Application

In the special case when $|D| = 2$, the results above yield a very simple derivation of the result obtained by Schaefer for the GENERALISED SATISFIABILITY problem [15]. This problem is described in the following example.

Example 6.1 (*GENERALISED SATISFIABILITY*). Let $S = \langle D, R_i(i \in I)\rangle$ be a relational structure with universe $D = \{0, 1\}$.

An instance of the GENERALISED SATISFIABILITY problem over S [15, 4] consists of a formula, $\mathscr{F}$, in propositional logic, which is a conjunction of terms of the form $R_i(x_1, x_2, \ldots, x_{\rho(i)})$, where the x_j are propositional variables. The question is whether there is an assignment of truth values to the variables in $\mathscr{F}$ such that $\mathscr{F}$ is true.

This can be expressed as the GCP instance $\langle\langle V, E_i(i \in I')\rangle, \langle D, R_i(i \in I')\rangle\rangle$, where V is the set of propositional variables used in $\mathscr{F}$, $I' \subseteq I$ is the set of indices, i, such that R_i appears in $\mathscr{F}$, and each E_i contains those tuples $\langle x_1, x_2, \ldots, x_{\rho(i)}\rangle$ such that $R_i(x_1, x_2, \ldots, x_{\rho(i)})$ is a conjunct of $\mathscr{F}$.

The results above allow us to describe completely the possible tractable subproblems of this problem.

Corollary 6.2. *For any set,* Γ, *of Boolean relations, if at least one of the following conditions holds*:

1. *Every relation in* Γ *holds when all variables are False.*
2. *Every relation in* Γ *holds when all variables are True.*
3. *Every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct has at most one negated variable.*
4. *Every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct has at most one unnegated variable.*
5. *Every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct contains at most two literals.*
6. *Every relation in* Γ *is the set of solutions of a system of linear equations over the finite field* $GF(2)$.

then GCP(Γ) *is in* **P**. *Otherwise* GCP(Γ) *is* **NP**-*complete.*

Proof. When $D=\{True,False\}$ there are just two possible constant operations, two idempotent binary operations (conjunction and disjunction), one majority operation, one generalised parity operation and no semiprojections. In order to be closed under one of these operations, all the relations in Γ must satisfy one of the properties listed [7] and this means that GCP(Γ) is in **P**, by the results above.

If Γ is not closed under any of these operations, then $\Gamma^{\triangleright}$ must contain essentially unary operations only, by Theorem 4.12, so GCP(Γ) is **NP**-complete, by Proposition 5.6. □

The next result gives a more uniform statement of the criterion for tractability, using Proposition 5.8.

Corollary 6.3. *For any finite set, Γ, of Boolean relations, let P be the* GCP *instance* $\langle S^3,S\rangle$, *where* $S=\langle\{0,1\},\Gamma\rangle$.

If all solutions to P are essentially unary, then GCP(Γ) *is* **NP**-*complete, otherwise* GCP(Γ) *is in* **P**.

This result shows that we can test any given finite set of Boolean relations, to establish whether or not it falls into one of the tractable classes, simply by solving a particular instance of the GENERALISED SATISFIABILITY problem involving the given relations and containing just eight Boolean variables. This provides a remarkably straightforward answer to the question raised by Schaefer in 1978, as to whether any such test exists [15].

7. Conclusions

The results presented in this paper lay the foundation for a novel algebraic theory of complexity in combinatorial problems.

We have established a strong link between the study of algebraic closure operations and the study of computational complexity. We have also demonstrated the application of the results derived in this paper to an important special case.

These results also give considerable insight into more general cases, when $|D|>2$. For example, it was shown in [6] that all the classes of tractable constraints which have been identified for the CONSTRAINT SATISFACTION problem (Example 2.20) are characterised by simple algebraic closure properties.

We expect the link between algebraic properties of relations and computational complexity described above to lead to considerable further progress in understanding the boundary between tractable and intractable combinatorial problems. For example, this work may provide a useful approach to combinatorial problems whose complexity is not yet fully characterised, such as GRAPH HOMOMORPHISM to a fixed directed graph [1].

References

[1] J. Bang-Jensen, P. Hell, G. MacGillivray, On the complexity of colouring by superdigraphs of bipartite graphs, Discrete Math. 109 (1992) 27–44.
[2] P.M. Cohn, Universal Algebra, Harper & Row, New York, 1965.
[3] M.C. Cooper, D.A. Cohen, P.G. Jeavons, Characterizing tractable constraints, Artificial Intelligence 65 (1994) 347–361.
[4] M.R. Garey, D.S. Johnson, Computers and intractability: a Guide to NP-Completeness, Freeman, San Francisco, CA, 1979.
[5] D. Geiger, Closed systems of functions and predicates, Pacific J. Math. 27 (1968) 95–100.
[6] P.G. Jeavons, D.A. Cohen, M. Gyssens, A unifying framework for tractable constraints, U. Montanari, F. Rossi (Eds.), Constraint Programming CP'95, Lecture Notes in Computer Science, vol. 976, 1995, pp. 276–291.
[7] P. Jeavons, D. Cohen, An algebraic characterization of tractable constraints, D.-Z. Du, M. Li (Eds.), Computing and Combinatorics COCOON '95, Lecture Notes in Computer Science, vol. 959, 1995, pp. 633–642.
[8] S. Kasif, On the parallel complexity of discrete relaxation in constraint satisfaction networks, Artificial Intell. 45 (1990) 275–286.
[9] L. Kirousis, Fast parallel constraint satisfaction, Artificial Intell. 64 (1993) 147–160.
[10] P.B. Ladkin, R.D. Maddux, On binary constraint problems, J. ACM 41 (1994) 435–469.
[11] R.N. McKenzie, G.F. McNulty, W.F. Taylor, Algebras, Lattices and Varieties, Vol. I, Wadsworth and Brooks, California, 1987.
[12] U. Montanari, Networks of constraints: fundamental properties and applications to picture processing, Inform. Sci. 7 (1974) 95–132.
[13] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, MA, 1994.
[14] I.G. Rosenberg, Minimal clones I: the five types, in: Lectures in Universal Algebra (Proc. Conf. Szeged 1983), Colloq. Math. Soc. Janos Bolyai 43, North-Holland, Amsterdam, 1986, pp. 405–427.
[15] T.J. Schaefer, The complexity of satisfiability problems, Proc. 10th ACM Symp. on Theory of Computing (STOC), 1978, pp. 216–226.
[16] A. Szendrei, Clones in Universal Algebra, Seminaires de Mathematiques Superieures 99, University of Montreal, 1986.