Electronic Notes in Theoretical Computer Science 33 (2000)
URL: http://www.elsevier.nl/locate/entcs/volume33.html 10 pages

Constraints, Adjunctions and (Co)algebras

Ola Angelsmark

Department of Computer and Information Science
Linköpings universitet
S-581 83 Linköping, Sweden
email: olaan@ida.liu.se

Abstract

The connection between constraints and universal algebra has been looked at in, *e.g.*, Jeavons, Cohen and Pearson, 1998, and has given interesting results. Since the connection between universal algebra and category theory is so obvious, we will in this paper investigate if the usage of category theory has any impact on the results and/or reasoning and if anything can be gained from this approach.

We construct categories of problem instances and of solutions to these, and, via an adjunction between these categories, note that the algebras give us a way of describing 'minimality of a problem,' while the coalgebras give a criterion for deciding if a given set of solutions can be expressed by a constraint problem of a given arity.

Another pair of categories, of sets of relations and of sets of operations on a fixed set, is defined, and this time the algebras we get give an indication of the 'expressive power' of a set of constraint types, while the coalgebras tell us about the computational complexity of the corresponding constraint problem.

1 Introduction

The constraint satisfaction problem was first formulated by Montanari in 1974 [4] when he used it as a way of describing certain combinatorial problems arising in image processing. Fairly soon it was realised that the framework was useful in a much broader class of problems and it has since been the subject of intense research, theoretical as well as experimental.

Intuitively, a constraint satisfaction problem aims at, given a set of variables subject to certain constraints on the values they can assume, finding an assignment of values such that no constraint is violated.

A classical example of a problem often formulated as a constraint satisfaction problem is the n-queens problem. Given n queens, place them on an $n \times n$ squares chess board in such a way that no queen threatens (can capture) any other queen.

Another example is the problem of sceduling a collection of tasks. Given the tasks and a set of constraints on them, *e.g.*, which tasks can be performed simultaneously, which has to preceed wich others, etc., find an assignment of times these tasks are carried out such that no constraint is violated.

Other examples include several classical combinatorial problems, such as the satisfiability problem from propositional logic and the colorability problem from graph theory, which can quite naturally be expressed as constraint satisfaction problems.

The *complexity* of constraint satisfaction problems has also been the subject of intense research. Finding a solution by brute-force methods, *i.e.*, going through all possible assignments and check for constraint violations, is generally not an option. The reader is referred to Pearson & Jeavons [6] for an in-depth discussion of the complexity of constraint satisfaction problems.

The link between universal algebra and constraint satisfaction problems has been explored, *e.g.*, in Jeavons, Cohen & Pearson [2], and turned out to be fruitful. In this paper we will translate this to category theory and thus look at the connection between category theory and constraint satisfaction problems. Though certainly no groundbreaking discoveries are made, the first steps towards applying more advanced categorical techniques to the problem are taken.

1.1 Overview of the paper

We begin with defining the constraint satisfaction problem and then construct two categories consisting of problem instances and solutions to these, respectively. We proceed with defining a pair of functors between them, and note that these form an adjunction. The algebras given by the corresponding monad give us a way of describing the property of 'minimality of a problem', while the coalgebras give a criterion for deciding if a given set of solutions can be expressed by a constraint problem of a given arity. From this we move on to define another pair of categories, this time they consist of sets of relations and of sets of operations on a fixed set. An adjoint pair of functors between these categories is defined, and the monads and comonads the pair give rise to hand us algebras that give an indication of the 'expressive power' of a set of constraint types, and coalgebras which tells us about the computational complexity of the corresponding constraint problem. We then conclude the discussion with a few suggestions on future work.

2 Relations and constraints

Since definitions and notations varies slightly between authors, we define the basic notions to make sure we agree on them.

First of all, we need to define what we mean by *a relation over D*. 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.

A constraint satisfaction problem is a triple $\langle V, D, C\rangle$ where:

- V is a set of variables
- D is a domain of values
- C is a set of constraints $\{c_1, c_2, \ldots, c_q\}$

Each constraint $c_i \in C$ is a pair $\langle s_i, R_i\rangle$ where:

- s_i is a tuple of length m_i of variables, called the constraint scope
- R_i is an m_i-ary relation over D, the constraint relation

A *solution* to a constraint satisfaction problem instance is a function $f : V \to D$ 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.

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.

The *rank function* of a relational structure $\langle V, E_1, E_2, \ldots, E_k\rangle$ is a function $\rho : \{1, 2, \ldots, k\} \to \mathbb{N}$ 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' iff they have identical rank functions.

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 $h : V \to V'$ s.t. 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 $\mathrm{Hom}(\Sigma, \Sigma')$.

Proposition 2.1 *For any constraint satisfaction problem $\mathcal{P} = \langle V, D, C\rangle$ with $C = \{\langle s_1, R_1\rangle, \langle s_2, R_2\rangle, \ldots, \langle s_q, R_q\rangle\}$, 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 $\Sigma' = \langle D, R_1, R_2, \ldots, R_q\rangle$.*

Proof. See Jeavons, Cohen & Pearson [2]. □

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'

We also need an ordering on the relational structures, which the following definition gives us:

Definition 2.2 Let $\Sigma = \langle D, E_1, E_2, \ldots, E_k\rangle$, $\Sigma' = \langle D, E'_1, E'_2, \ldots, E'_k\rangle$ be similar relational structures. If $E_i \supseteq E'_i$ for all $i \in \{1, 2, \ldots, k\}$, then we say that $\Sigma \sqsubseteq \Sigma'$.

3 The categories $\Delta_{D,\rho}$ and $\wp(\mathrm{D}^{\mathrm{V}})$

In the following section we have chosen a fixed relational structure Σ_0 with universe V and rank function ρ together with a fixed set D. We now consider all instances $\langle\Sigma_0, \Sigma\rangle$ where Σ varies over all relational structures with universe D and rank function ρ.

Define a category of relational structures as follows:

Definition 3.1 Define $\Delta_{D,\rho}$ as having

- relational structures Σ with universe D and rank function ρ as objects, and
- there is an arrow $\Sigma \to \Sigma'$ iff $\Sigma \sqsubseteq \Sigma'$.

Definition 3.2 Define the category $\wp(\mathrm{D}^{\mathrm{V}})$ of mappings $V \to D$ as follows:

- The objects of $\wp(\mathrm{D}^{\mathrm{V}})$ are sets of mappings $V \to D$, and
- there is an arrow $M \to M'$ in $\mathrm{ar}(\wp(\mathrm{D}^{\mathrm{V}}))$ iff $M \subseteq M'$.

These categories are both partial order categories and thus have a number of interesting properties we will make use of later.

3.1 The functors $\mathrm{Sol}_{\Sigma_0}(-)$ *and* $\mathrm{Pro}_{\Sigma_0}(-)$

Let $\Sigma_0 = \langle V, E_1, E_2, \ldots, E_k\rangle$ be a fixed relational structure with rank function ρ and define two functors, $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$ as:

Definition 3.3 Define the functor $\mathrm{Sol}_{\Sigma_0}(-) : \Delta_{D,\rho} \longrightarrow \wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}$ as mapping

- objects $\Sigma \in \mathrm{ob}(\Delta_{D,\rho})$ to the set $\mathrm{Hom}(\Sigma_0, \Sigma)$ of homomorphisms from Σ_0 to Σ, and
- arrows $f : \Sigma \to \Sigma'$ to the arrow given by the subset relation in $\wp(\mathrm{D}^{\mathrm{V}})$, reversing the direction.

Note that since $\mathrm{Sol}_{\Sigma_0}(-)$ reverses the direction of the arrows, we get that if there is an arrow $f : \Sigma \to \Sigma'$ in $\Delta_{D,\rho}$ then $\mathrm{Hom}(\Sigma_0, \Sigma) \subseteq \mathrm{Hom}(\Sigma_0, \Sigma')$, so f is mapped to the arrow $\mathrm{Hom}(\Sigma_0, \Sigma) \leftarrow \mathrm{Hom}(\Sigma_0, \Sigma')$ in $\wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}$.

Definition 3.4 Now define $\mathrm{Pro}_{\Sigma_0}(-) : \wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}} \longrightarrow \Delta_{D,\rho}$ as working on

- objects by mapping an object $M \in \mathrm{ob}(\wp(\mathrm{D}^{\mathrm{V}}))$ to a relational structure $\mathrm{Pro}_{\Sigma_0}(M) = \langle D, R_1, R_2, \ldots, R_k\rangle$ where

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

  while
- arrows $f : M \to M'$ are mapped as expected.

That the functors satisfy the functor axioms follows from the properties of the underlying relations.

Intuitively, $\mathrm{Sol}_{\Sigma_0}(-)$ maps a relational structure Σ to the set of solutions to $\langle\Sigma_0,\Sigma\rangle$ while $\mathrm{Pro}_{\Sigma_0}(-)$ can be thought of as given a set of mappings 'constructing' a relational structure Σ such that the mappings constitute solutions to the problem instance $\langle\Sigma_0,\Sigma\rangle$.

To show that the functors form an adjoint pair we will need the following theorem:

Theorem 3.5 (Galois connections are adjoint pairs) *Let $\mathscr{P},\mathscr{Q}$ be two preorder categories and $F:\mathscr{P}\longrightarrow\mathscr{Q}^{op}, G:\mathscr{Q}^{op}\longrightarrow\mathscr{P}$ two order-preserving functions (regarded as functors.) Then F is left adjoint to G iff, for all $p\in\mathrm{ob}(\mathscr{P})$ and $q\in\mathrm{ob}(\mathscr{Q})$,*

$$F(p)\geq q \text{ in } \mathscr{Q} \Leftrightarrow p\leq G(q) \text{ in } \mathscr{P}$$

When this is the case, there is exactly one adjunction φ making F the left adjoint for G. For all p and q, $p\leq GF(p)$ and $FG(q)\geq q$; hence also

$$F(p)\geq FGF(p)\geq F(p),\quad G(q)\leq GFG(q)\leq G(q)$$

Proof. See MacLane [3]. □

The unit and counit of the adjunction are given by $FG(q)\geq q$ and $GF(p)\geq p$ for all p and q.

When the relations in question are antisymmetric as well as reflexive and transitive (*i.e.*, we have a partial ordering), the last line of the proposition above collapses to

$$FGF(p)=F(p),\quad GFG(q)=G(q)$$

Proposition 3.6 *The functors $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$ as defined above form an adjoint pair with $\mathrm{Sol}_{\Sigma_0}(-)$ left adjoint to $\mathrm{Pro}_{\Sigma_0}(-)$.*

Proof. By theorem 3.5 we have to show

$$\mathrm{Sol}_{\Sigma_0}(\Sigma)\leftarrow M \text{ in } \wp(\mathrm{D}^{\mathrm{V}}) \Leftrightarrow \Sigma\rightarrow\mathrm{Pro}_{\Sigma_0}(M) \text{ in } \Delta_{D,\rho}$$

Assume there is an arrow $M\rightarrow\mathrm{Sol}_{\Sigma_0}(\Sigma)$ in $\wp(\mathrm{D}^{\mathrm{V}})$, that is, the solution to the problem instance $\langle\Sigma_0,\Sigma\rangle$ is contained in M. From the definition of $\mathrm{Pro}_{\Sigma_0}(-)$ it then follows that $\Sigma\sqsubseteq\mathrm{Pro}_{\Sigma_0}(M)$, *i.e.*, there is an arrow $\Sigma\rightarrow\mathrm{Pro}_{\Sigma_0}(M)$ in $\Delta_{D,\rho}$.

Analogously, if we assume the existence of an arrow $\Sigma\rightarrow\mathrm{Pro}_{\Sigma_0}(M)$ in $\wp(\mathrm{D}^{\mathrm{V}})$, then it follows that the set of solutions to $\langle\Sigma_0,\Sigma\rangle$ is a subset of M. □

The adjunction we get is $\langle\mathrm{Sol}_{\Sigma_0}(-),\mathrm{Pro}_{\Sigma_0}(-),\eta,\varepsilon\rangle$ with unit $\eta=\Sigma\rightarrow\mathrm{Pro}_{\Sigma_0}(\mathrm{Sol}_{\Sigma_0}(\Sigma))$ and counit $\varepsilon=\mathrm{Sol}_{\Sigma_0}(\mathrm{Pro}_{\Sigma_0}(M))\rightarrow M$. This adjunction gives us a monad in the category $\Delta_{D,\rho}$ which, since the category is a partial order, is a functor $PS:\Delta_{D,\rho}\longrightarrow\Delta_{D,\rho}$[1] satisfying $\Sigma\sqsubseteq PS(\Sigma)$ and $PS(PS(\Sigma))=PS(\Sigma)$ for all Σ in $\Delta_{D,\rho}$. A PS-algebra is then an element Σ with $PS(\Sigma)=\Sigma$.

Dually, we get a comonad in $\wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}$, a functor $SP:\wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}\longrightarrow\wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}$ satisfying $M\subseteq SP(M)$ (*i.e.*, there is an arrow $SP(M)\rightarrow M$ in $\wp(\mathrm{D}^{\mathrm{V}})^{\mathrm{op}}$) and

[1] PS is of course an abbreviation of $\mathrm{Pro}_{\Sigma_0}(\mathrm{Sol}_{\Sigma_0}(-))$

$SP(SP(M)) = SP(M)$. An SP-coalgebra is an element M with $SP(M) = M$.

Since the categories are partial orderings, we can deduce from theorem 3.5 that the following identities hold:

$$\mathrm{Sol}_{\Sigma_0}(\mathrm{Pro}_{\Sigma_0}(\mathrm{Sol}_{\Sigma_0}(\Sigma))) = \mathrm{Sol}_{\Sigma_0}(\Sigma)$$

and

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

Consequently, any element M of $\wp(\mathrm{D}^{\mathrm{V}})$ is mapped to a PS-algebra in $\Delta_{D,\rho}$ and any element Σ in $\Delta_{D,\rho}$ is mapped to an SP-coalgebra in $\wp(\mathrm{D}^{\mathrm{V}})$.

3.2 PS-algebras and SP-coalgebras

Let us now have a look at what the algebras and coalgebras in the previous section can offer us.

Proposition 3.7 *Let $K_n = \langle V, \{e_1, e_2, \ldots\}\rangle$ be a complete graph with n vertices, and set $\Sigma_{K_n} = \langle V, \{e_1\}, \{e_2\}, \ldots\rangle$.*

*For any **GCP** instance $\mathcal{P} = \langle \Sigma_{K_n}, \Sigma'\rangle$, a minimal binary constraint satisfaction problem, as defined in Montanari [4], with the same solution as $\mathcal{P}$ is given by*

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

Hence $\mathcal{P}$ is a minimal binary satisfaction problem iff

$$\mathrm{Pro}_{\Sigma_{K_n}}(\mathrm{Sol}_{\Sigma_{K_n}}(\Sigma)) = \Sigma.$$

Proof. See Jeavons, Cohen & Pearson [2]. □

The problem of deriving the unique minimal constraint network with the same solution as a given problem instance is described by Montanari as the central problem in many practical applications [4], so it is clear that the PS-algebras are of interest.

Proposition 3.8 *A set of mappings M in $\wp(\mathrm{D}^{\mathrm{V}})$ is the set of solutions to some constraint satisfaction problem instance with constraint hypergraph Σ_0 iff*

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

Proof. See Jeavons, Cohen & Pearson [2]. □

Thus the problem instance $\mathcal{P}$ in proposition 3.7 is a minimal binary satisfaction problem iff Σ is a PS-algebra and a set M of mappings is the set of solutions to some constraint satisfaction problem instance iff M is an SP-coalgebra.

4 The categories Λ_D and Ω_D

We will now consider the collection of instances of the generalised constraint satisfaction problem where we fix the domain of values and the set of possible constraint relations and let the hypergraphs vary. Let us start with defining two new categories, the categories of relations and of operations on a fixed domain, respectively.

Definition 4.1 Define the categories Ω_D and Λ_D as follows:

- Let Λ_D be the category of all sets of relations on a given set D, ordered by inclusion, and
- let Ω_D be the category of all sets of operations on D, ordered by inclusion.

The following definitions are needed in the construction of the adjunction between these categories.

Definition 4.2 Let r be a relation over a given set D. An operation $o : D^n \to D$ is called a *polymorphism* of r iff $\forall t_1, t_2, \ldots, t_n \in r . o(t_1, t_2, \ldots, t_n) \in r$. If o is a polymorphism of r, r is said to be an *invariant* for o.

4.1 The functors Inv(−) *and* Pol(−)

The following two functors are well-known in universal algebra, see, *e.g.*, Rosenberg [7].

Definition 4.3 Define the functor $\mathrm{Pol} : \Lambda_D \longrightarrow \Omega_D^{\mathrm{op}}$ as follows:

- For every object R in Λ_D,

$$\mathrm{Pol}(R) = \{o \mid \forall r \in R, o \text{ is a polymorphism of } r\}$$

  and
- an arrow $R \to R'$ in Λ_D is mapped to the arrow $\mathrm{Pol}(R) \leftarrow \mathrm{Pol}(R')$ in Ω_D^{op}.

That Pol(−) really is a functor follows from the following argument:

Suppose there is an arrow $R \to R'$ in Λ_D, *i.e.*, $R \subseteq R'$. Any operation o which is a polymorphism of R' is also a polymorphism of R, since $R \subseteq R'$, so $\mathrm{Pol}(R') \subseteq \mathrm{Pol}(R)$ and thus there is an arrow $\mathrm{Pol}(R') \leftarrow \mathrm{Pol}(R)$ in Ω_D^{op}.

Definition 4.4 Now define the functore $\mathrm{Inv}(-) : \Omega_D^{\mathrm{op}} \longrightarrow \Lambda_D$ as

- mapping an object O in Ω_D to the set $\{r \mid \forall o \in O . r \text{ is an invariant for } o\}$ and
- an arrow $O \to O'$ in Ω_D^{op} is mapped to the arrow $\mathrm{Inv}(O) \leftarrow \mathrm{Inv}(O')$ in Λ_D.

Inv(−) is a functor, which we can convince ourselves of by noting that if there is an arrow $O \leftarrow O'$ in Ω_D^{op}, *i.e.*, O is a subset of O', then given any relation r in $\mathrm{Inv}(O')$ and an operation o in O, this operation must also be in O', so r is an invariant for O and thus $r \in \mathrm{Inv}(O)$.

4.2 Inv(−) *and* Pol(−) *form an adjoint pair*

Similarly to the functors $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$, Inv(−) and Pol(−) form an adjoint pair, but before we show this, we note the following:

Lemma 4.5

(i) *For any object* R *in* Λ_D*, the arrow* $R \to \mathrm{Inv}(\mathrm{Pol}(R))$ *exists in* $\mathrm{ar}(\Lambda_D)$*.*

(ii) *For any object* O *in* Ω_D*, the arrow* $O \to \mathrm{Pol}(\mathrm{Inv}(O))$ *exists in* $\mathrm{ar}(\Omega_D^{op})$*.*

Proof.

(i) Let r be a relation on D and consider $\mathrm{Inv}(\mathrm{Pol}(\{r\}))$. By the definition of Pol(−) we get that

$$\mathrm{Inv}(\mathrm{Pol}(\{r\})) = \mathrm{Inv}(\{o \mid o \text{ polymorphism of } r\}) =$$
$$\{r' \mid o' \in \{o \mid o \text{ polymorphism of } r\} \wedge r' \text{ invariant for } o'\}$$

Clearly, r belongs to this set.

(ii) Now let o be an operation on D and consider $\mathrm{Pol}(\mathrm{Inv}(\{o\}))$. From the definition of Inv(−) we get

$$\mathrm{Pol}(\mathrm{Inv}(\{o\})) = \mathrm{Pol}(\{r \mid r \text{ invariant for } o\}) =$$
$$\{o' \mid r' \in \{r \mid r \text{ invariant for } o\} \wedge o' \text{ polymorphism of } r'\}$$

And again we see that o must belong to this set.

□

Proposition 4.6 *The functors* $\mathrm{Pol}(-) : \Lambda_D \longrightarrow \Omega_D^{op}$ *and* $\mathrm{Inv}(-) : \Omega_D^{op} \longrightarrow \Lambda_D$ *form an adjoint pair with* Pol(−) *left adjoint to* Inv(−)*.*

Proof. As was the case with $\mathrm{Sol}_{\Sigma_0}(-)$ and $\mathrm{Pro}_{\Sigma_0}(-)$, by theorem 3.5 we have to show the following:

$$\mathrm{Pol}(R) \leftarrow O \text{ in } \Omega_D^{\mathrm{op}} \Leftrightarrow R \to \mathrm{Inv}(O) \text{ in } \Lambda_D$$

First, assume $\mathrm{Pol}(R) \leftarrow O$, so $\mathrm{Inv}(\mathrm{Pol}(R)) \to \mathrm{Inv}(O)$ and since by lemma 4.5 we know that there is an arrow $R \to \mathrm{Inv}(\mathrm{Pol}(R))$, composition of these gives us the arrow $R \to \mathrm{Inv}(O)$. Similarly, given $R \to \mathrm{Inv}(O)$, $\mathrm{Pol}(R) \leftarrow \mathrm{Pol}(\mathrm{Inv}(O))$ and since the arrow $O \to \mathrm{Pol}(\mathrm{Inv}(O))$ exists, composition gives us that $\mathrm{Pol}(R) \leftarrow O$ exists. □

The adjunction given by this pair, $\langle \mathrm{Pol}(-), \mathrm{Inv}(-), \eta, \varepsilon \rangle$, has unit $\eta : R \to \mathrm{Inv}(\mathrm{Pol}(R))$ and counit $\varepsilon : O \to \mathrm{Inv}(\mathrm{Pol}(O))$. This gives us a monad on the category Λ_D which is a functor $IP : \Lambda_D \longrightarrow \Lambda_D$ satisfying $R \subseteq IP(R)$ and $IP(IP(R)) = IP(R)$ for every object $R \in \Lambda_D$. An IP-algebra is then an element R in Λ_D with $IP(R) = R$. We also get a comonad on Ω_D^{op}, a functor $PI : \Omega_D^{\mathrm{op}} \longrightarrow \Omega_D^{\mathrm{op}}$ with $O \subseteq PI(O)$ and $PI(PI(O)) = PI(O)$, so a PI-coalgebra is an element O of Ω_D with $PI(O) = O$.

4.3 IP-algebras and PI-coalgebras

In Jeavons, Cohen & Pearson [2], the significance of the $\text{Inv}(\text{Pol}(-))$ functor is shown. Intuitively, the set of relations $\text{Inv}(\text{Pol}(R))$ is exactly those relations which can be expressed by problem instances in **GCP**(R), where **GCP**(R) is the collection of instances $\langle \Sigma, \Sigma' \rangle$ of generalised combinatorial problems where every relation in Σ' is an element of R.

Since $\text{Pol}(\text{Inv}(\text{Pol}(R))) = \text{Pol}(R)$, it stands to reason that every object R in Λ_D is mapped to a PI-coalgebra. In Jeavons [1], is shown that $\text{Pol}(R)$ determines (up to polynomial-time reductions) the complexity of R. By showing that $\text{Pol}(R)$ has to satisfy at least one out of six conditions, the possible choices of $\text{Pol}(R)$ is reduced, and each one of the conditions is associated with a complexity class.

5 Conclusion

The complexity of constraint satisfaction problems is an important issue and certainly worthy of the attention it has received in the past. By formulating the problem in this setting it is hoped that further analysis, using results from category theory, is simplified and it is quite possible much work already done in this field can be adapted more or less easily.

5.1 Future Work

Future work naturally includes further examination of the (co-)algebras presented in this paper. Further analysis of the minimality characterisation given by the PS-algebras would certainly be interesting, and, due to the importance of computational complexity, the PI-coalgebras deserve some attention.

As was noted by one of the anonymous referees of this paper, studying categories with more structure than a mere ordering could be of interest, say, categories with morphisms expressing the dynamics of partially solving constraint problems. The suggested reference, Mukai [5], looks very interesting.

Linköping, March, 2000

References

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

[2] P. G. Jeavons, D. A. Cohen, and J. K. Pearson. Constraints and universal algebra. *Annals of Mathematics and Artificial Intelligence*, 24:51–67, 1998.

[3] S. MacLane. *Categories for the Working Mathematician.* Graduate Texts in Mathematics. Springer-Verlag, New York, 1971.

[4] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

[5] K. Mukai. CLP(AFA): Coinductive semantics of Horn clauses with compact constraints. In J. Barwise, J. M. Gawron, G. Plotkin, and S. Tutiya, editors, *Situation Theory and Its Applications (Vol. 2)*, pages 179–214. CSLI, Stanford, CA, 1991.

[6] J. K. Pearson and P. G. Jeavons. A survey of tractable constraint satisfaction problems. Technical Report CSD-TR-97-15, Royal Holloway, University of London, July 1997.

[7] I. G. Rosenberg. Minimal clones I: The five types. In L. Szabó and Á. Szendrei, editors, *Lecture notes in Universal Algebra*, volume 43 of *Colloquia Mathematica Societatis János Bolyai*, 1983.