A complete classification of Constraint Languages generated by binary entropic minimal clones

Justin Pearson*

September 1, 2006

1 Preamble

While finishing this paper I found the paper [9] which proves that a binary operation that is generates a minimal clone is NP-complete is the operation is non-commutative and is polynomial if the operation is equivalent to a commutative operation that generates the clone. From this result all results in this paper can be (even more) trivially proved. This paper might be of interest to some readers as an introduction to the theory of the complexity of constraint satisfaction problems.

2 Introduction

A *Constraint Satisfaction Problems*(CSP)[25, 33] is a collection of *variables* which are to be assigned values from a *domain* and a collection of (often mutually conflicting) *constraints* that restrict the possible combinations of values that the variables can take. A solution is then a mapping from variables to the domain in which every constraint is satisfied. Graph colouring can be seen as a CSP: variables correspond to the nodes of the graph the domain is the set of colours that the graph is coloured with and between each pair of variables representing two nodes that are connected there is a constraint, $\neq$, forcing the two variables to take different values. Using different constraints (and non-binary constraints) more complicated problems can be expressed.[1] Practically [38, 37] many techniques have been developed: notably propagation [3, 31] and the use of specialised global constraints [1] which allow non-trivial problems to be solved in times that are often comparative with Integer linear programming [35].

*`mailto:justin.pearson@it.uu.se`

[1]Although all NP-complete problems are polynomially equivalent, so that for example boolean satisfiability can be expressed as a graph colouring problem. Part of the power of CSPs is that often the structure of a problem can be captured in a more natural way without coding the structure away.

The general Constraint satisfaction Problem (CSP) is NP complete [18]. It is a natural to ask what restrictions on can be put on the general CSP to guarantee that problem instances can be solved in polynomial time. There are in general two ways of putting restrictions on a CSP: either structural restrictions[19, 12] on how constraints can be put together or restrictions on which constraints can be used[11]. For example in graph colouring a structural restriction would correspond to limiting the types of graphs that are allowed to be coloured; for example, trees can be coloured in polynomial time. While limiting the types of constraints that are allowed in the graph colouring would corresponding to changing the $\neq$ constraints; for example, restricting the colours to only two results in a tractable sub-class of graph colouring.

In this paper we shall look at what restrictions can be placed on which constraint are used to build up CSP so that all instances are tractable. By a *constraint language* we simply mean a set of constraints: a CSP instance belongs to a constraint language if it is only built with constraints from that language. There has been a long line of investigation into tractable constraint languages[17, 24, 20, 27, 16, 11, 7, 6]. Of particular interest are languages that are maximally tractable: in that when *any* other new constraint is added to the language the language becomes NP-complete. These maximally tractable languages correspond to mathematical structures known as minimal clones [36, 32, 23, 30, 13].

3 Definitions

We recall some notations that are used in [6, 29] as well as some notions from universal algebra[36, 26, 10].

3.1 The Constraint Satisfaction Problem

Denote the set of n-tuples of a set D as D^n.

Definition 1. *An instance of a* constraint satisfaction problem *consists of the following:*

- *a finite set of variables, V;*
- *a finite domain of values, D;*
- *a set of constraints $\{C_1, \ldots, C_q\}$ where each constraint is a pair (S_i, R_i) where S_i is a list of variables of length m_i called the* constraint scope *and R_i is a subset of D^{m_i} called the* constraint relation.

A *solution* to an instance of the constraint satisfaction problem is a function, $f : V \to D$, from the set of variables to the domain such that the image of each constraint scope is an element of the corresponding relation that is for all i, $f(S_i) \in R_i$.

Definition 2. *Given a set of relations,* Γ*, define* $\mathrm{CSP}(\Gamma)$ *to be the class of decisions problems in which* instances *are CSPs in which all constraint relations come from* Γ*.*

The classification problem is, for which sets Γ are all problems in $\mathrm{CSP}(\Gamma)$ tractable? For the case of the 2-element domain was classified in [34] although many special cases for arbitrary domains [20, 27, 15, 29, 6] only recently has a full classification for the 3-element domain been achieved [5].

3.2 Functions and clones

Given an n-ary functions $f : D^n \to D$ and n m-ary functions, $g_1, \ldots, g_n$, the composition is defined to be

$$f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m)).$$

A set of functions is called a *clone* if it contains all projections and is closed under composition. A clone is *trivial* if it only contains projections. A clone is a called is called *minimal* if every proper sub-clone is a trivial clone.

Given a set a n-ary function $f : D^n \to D$ we say that f *preserves* an m-ary relation R if for every list (not necessarily distinct) of n-tuples $(d_{11}, \ldots, d_{m1}), \ldots, (d_{1n}, \ldots, d_{mn})$ belonging to R we have that:

$$f\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} = \begin{pmatrix} f(a_{11}, a_{12}, \ldots, a_{1n}) \\ f(a_{21}, a_{22}, \ldots, a_{2n}) \\ \vdots \\ f(a_{m1}, a_{m2}, \ldots, a_{mn}) \end{pmatrix} \in R.$$

Given a set of functions C the set of all relations that are preserved by all members of C is denoted $Inv(C)$ and given a set of relations F the set of functions preserving all members of F is denoted $Pol(F)$.

The two functions Inv and Pol form a Galois connection. The closed sets of functions $Pol(Inv(C))$ are clones. The closed sets of relations $Inv(Pol(F))$ are known as relation clones[36, 28]. The following theorem states that the interesting sets of relations to study for tractability are exactly sets of relations of the form $Pol(Inv(C))$.

Theorem 1. *[20] Let* Γ_0 *and* Γ *be sets of relations over a finite set* D *where* Γ *is finite. if* $\Gamma \subseteq Inv(Pol(\Gamma_0))$ *then* $\mathrm{CSP}(\Gamma)$ *is reducible to* $\mathrm{CSP}(\Gamma_0)$ *in polynomial time.*

Since Inv and Pol form a Galois connection the minimal clones correspond to sets of relations that are maximal. Where maximal means that adding any other relations generates the relational clone of all relations. This is because adding an extra relation, R, will give the clone $Inv(Pol(Inv(C) \cup \{R\}))$ which is a subclone of $Inv(C)$ which must be a trivial clone. Hence $Pol(Inv(C) \cup \{R\})$ must contain all relations giving an NP-complete language.

Given a function f the clone generated by f is the set of functions $Pol(Inv(\{f\}))$. We say a clone, C, is tractable if all *finite* subsets $\Gamma \in Inv(C)$ then the decision problem CSP(Γ) is tractable and C is NP-complete if there is a subset $\Gamma \in Inv(C)$ such that CSP(Γ) is NP-complete.

Minimal clones have been classified in [32], all types of minimal clones have been classified for tractability in [20] except for clones generated by a binary idempotent operation which contains both NP-complete clones as well as tractable clones. These clones where finally classified in [9].

3.3 Universal Algebra

In this section the necessary background from universal algebra is presented sufficiently to state the classification theorem for binary entropic clones from [23].

Universal algebra is the study of general algebraic structures. An algebra $\mathcal{A}$ is a pair $(A, (f_i)_{i \in I})$ where each f_i is a function from $D^{m_i} \to D$ for some m_i. When the algebra consists of a single operation the algebra will simply be written as a pair (A, f). In this paper we are only concerned with algebras $(A, \cdot)$ where $\cdot$ is binary operation. Instead of writing $\cdot(a, b) = c$ we shall write $a \cdot b = c$. Much of universal algebra consists in studying equations that algebras satisfy. The correspondence between classes of algebras and the equations that they satisfy is the theory of varieties[10].

Give an algebra $\mathcal{A} = (A, f)$ we say that $\mathcal{A}$ is tractable (or NP-complete) if the relational clone $Inv\{f\}$ is tractable (or NP-complete). Obviously this can be extended to algebras containing more than one operation details can be found in [11, 6].

Definition 3. *An algebra, $(A, \cdot)$ with a single binary operation is* entopric *if for all:*

$$\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix}$$

we have that

$$(x_1 \cdot x_2) \cdot (y_1 \cdot y_2) = (x_1 \cdot y_1) \cdot (x_2 \cdot y_2)$$

In fact the definition generalises to algebras with operations of arbitrary arity (see [23, 2]).

Definition 4. *A* rectangular band*[26] is an algebra satisfying the identity*

$$x \cdot (y \cdot z) = x \cdot z$$

It is proved in [27] then any rectangular band algebra is in NP.

Definition 5. *A binary operation is a* semilattice *if it is a commutative, idempotent and associative. That is $x \cdot x = x$,$x \cdot y = y \cdot x$ and $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ for all x, y and z.*

The fact that semilattices are tractable can be proved via variety of methods either as a generalisation of the proof in [22] or the problem can be proved [15] to have bounded width [16].

Definition 6. *A binary operation defines a* right semilattice *if it satisfies the entropic law and the following equations:*

1. $x \cdot x = x$
2. $x \cdot (x \cdot y) = x$
3. $(x \cdot y) \cdot y = x \cdot y$

Definition 7. *A binary operation is a p-cyclic groupoid if it is idempotent, satisfies the entopic laws and satisfies the equations:*

1. $x \cdot (x \cdot y) = x$
2. $\underbrace{(\cdots((x \cdot y) \cdot y) \cdots) \cdot y}_{p\ y\text{'s}} = x$

Definition 8. left normal band *is an idempotent algebra satisfying*

1. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
2. $x \cdot (y \cdot z) = x \cdot (z \cdot y)$

In this paper we are also interested in minimal clones generated by binary affine operations. That is given some finite field with a prime number of elements F_p then the algebras $(F_p, rx + (1-r)y)$ for $r \in F_p \setminus \{0, 1\}$ generates a minimal clone. It can be proved that any member $R \in Inv\{rx + (1-r)y\}$ is defined by a set of linear equations.

We can now state one of the the major theorems form [23].

Theorem 2. *If C is a entropic minimal clone generated by an idempotent binary operation then C is isomorphic or anti-isomorphic to the clone of one of the following classes of varieties:*

1. *Affine spaces of the form $(F_q, rx + (1-r)y)$,*
2. *Rectangular bands,*
3. *Semilattices,*
4. *right semilattices,*
5. *p-cycle groupoids,*
6. *Left normal bands.*

3.4 Proving complexity results from Algebra

The general strategy for proving NP-completeness is to show some NP-complete problem can be expressed in the constraint language. Tractability is proved by giving some algorithm (or class of algorithms) that solves any instance in the constraint language.

A function $f : D^n \to D$ is essentially unary if there exists a unary function g such that $f(a_1, \ldots, a_i, \ldots, a_n) = g(a_i)$ for some i and for all $a_1, \ldots, a_n \in D$.

Theorem 3. *[20] For any finite set D and for any set of relations Γ on D if $Pol(\Gamma)$ contains essentially unary operations* only *then* CSP(Γ) *is NP-complete.*

We shall only define the notion of sub-algebras for algebras with a single binary operation (for the more general definition see [10]).

Definition 9. *Given an algebra $\mathcal{A} = (A, \cdot)$ and a subset $U \subseteq A$ then $\mathcal{U} = (U, \cdot)$ is a subalgebra of $\mathcal{A}$ if for all $u, v \in U$ then $u \cdot v \in U$.*

It can be proved [6] that given an algebra $\mathcal{A}$ if $\mathcal{U}$ is a sub-algebra that is NP-complete then $\mathcal{A}$ is NP-complete (this is also true for homomorphic images, although we won't need that in this paper). For example given the algebra $(\{1, 2, 3\}, \cdot)$ with the Cayley table:

$\cdot$	1	2	3
1	1	2	2
2	1	2	1
3	3	2	3

then this algebra is NP-complete because the subalgebra on the elements $\{1, 2\}$ generates only essentially unary operations. The NP-completeness results in this paper consists in proving that certain classes of algebras always have two element sub-algebras that are essentially unary.

Most Tractability results can be divided into two distinct classes either tractability can be achieved by showing only local operations are needed, so called problems of bounded width[16, 15, 21] or showing that the problem has a group theoretic structure[16] (or more generally that there is a Malt'sev term in $Pol(\Gamma)$ see [4, 8]) which are non-local in nature. If the relations of a CSP are defined by sets of linear equations over finite fields then a solution to the CSP can be found in polynomial time by Gaussian elimination. Although there has recently been an interesting mixture of the two [14].

4 The complexity of Commutative Binary Minimal Clones

We have seen that affine spaces and semilattices are tractable. We now prove that all other non-commutative (and non affine) entropic minimal clones generated by a binary operation are NP-complete.

First of all a left-normal band which is commutative is a semilattice (by definition) and hence tractable.

Lemma 1. *Given a left normal band with elements α and β such that $\alpha \cdot \beta \neq \beta \cdot \alpha$ then the algebra is NP-complete.*

Proof. We have the following sub-algebra:

$\cdot$	$\alpha\beta$	$\beta\alpha$
$\alpha\beta$	$\alpha\beta$	$\alpha\beta$
$\beta\alpha$	$\beta\alpha$	$\beta\alpha$

The algebra is idempotent we just have to verify $(\alpha \cdot \beta) \cdot (\beta \cdot \alpha)$ and $(\beta \cdot \alpha) \cdot (\alpha \cdot \beta)$.

$$(\alpha \cdot \beta) \cdot (\beta \cdot \alpha) = (\alpha \cdot \beta) \cdot (\alpha \cdot \beta) = \alpha \cdot \beta$$

the other identity is verified in the same way. □

Hence any non-commutative left normal band has a two element algebra which is a projection (which is essentially unary) and hence it is NP-complete. We now prove that any right-semilattice or any p-cyclic groupoid is NP-complete.

Lemma 2. *An algebra satisfying the equations:*

1. $x \cdot x = x$
2. $x \cdot (x \cdot y) = x$
3. $x \cdot (y \cdot z) = x \cdot y$

is NP-complete.

Proof. We shall prove this by demonstrating that there is always a two-element sub-algebra which is a projection. Pick any element α there must be an element β such that $\alpha \cdot \beta = \alpha$. This because $x \cdot (x \cdot y) = x$ gives that $\alpha \cdot (\alpha \cdot \gamma) = \alpha$ for any γ. So setting $\beta = \alpha \cdot \gamma$ for some γ.

Further, we can pick γ such that $(\alpha \cdot \gamma) \neq \alpha$. For if not, suppose that $\alpha \cdot \beta = \alpha$ if and only if $\beta = \alpha$. Then for all $\delta_0, \dots \delta_n$ in the domain where $\delta_i \neq \alpha$ we have that:

$$\alpha \cdot (\alpha \cdot \delta_i) = \alpha$$

but since $(\alpha \cdot \delta_i) = \eta_i \neq \alpha$ we have that $\alpha \cdot \eta_i = \alpha$ giving a contradiction.

Hence we can pick β such that $\beta \neq \alpha$. Now given that $\alpha \cdot \gamma \neq \beta$ using the equation $x \cdot (y \cdot z) = x \cdot y$ to derive that $\beta \cdot \alpha = \beta \cdot (\alpha \cdot \gamma) = \beta \cdot \beta = \beta$ hence we have the following multiplication table for α and β:

$\cdot$	α	β
α	α	α
β	β	β

hence there is a two element sub-algebra with is a projection hence the algebra is NP-complete. □

Finally it is proved in [23] that any entropic binary algebra satisfying $x \cdot (x \cdot y) = x$ also satisfies $x \cdot (y \cdot z) = x \cdot y$ and further that is $\mathcal{V}$ is a variety of idempotent entropic algebra with a single binary function satisfying $x \cdot (x \cdot y) = x$ which has a minimal clone that $\mathcal{V}$ is either the variety of right semilattices or the variety of p-cyclic groupoids and is hence NP-complete by Lemma 2.

References

[1] Nicolas Beldiceanu, Mats Carlsson, and Jean-Xavier Rampon. Global constraint catalog. Technical Report T2005-08, Swedish Institute of Computer Science, November 2005.

[2] C. Bergman and J. Berman. Categorical equivalence of modes. *DISCUSSIONES MATHEMATICAE ALGEBRA AND STOCHASTIC METHODS*, 19(1):41–62, 1999.

[3] Christian Bessière and Jean-Charles Régin. Arc consistency for general constraint networks: Preliminary results. In *Proceedings of IJCAI'97*, pages 398–404, 1997.

[4] Andrei Bulatov. Malt'sev constraints are tractable. Technical Report PRG-RR-02-05, Oxford University Computing Laboratory,, 2002. http://web.comlab.ox.ac.uk/oucl/publications/tr/rr-02-05.html.

[5] Andrei Bulatov. A dichotomy theorem on a three element set. *Journal of the ACM*, 53(1):66–120, 2006.

[6] Andrei Bulatov, Krokhin Andrei, and Peter Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM journal on computing*, 34(3):720–742, 2005.

[7] Andrei Bulatov and Peter Jeavons. An algebraic approach to multi-sorted constraints. In *Proceedings of CP'03*, volume 2833 of *Lecture Notes in Computer Science*, pages 183–198. Springer Verlag, 2003.

[8] Andrei Bulatov and Dalmau Victor. A simple algorithm for Mal'tsev constraints. *Siam Jounral of Computing*, 36(1):16–27, 2006.

[9] Andrei A. Bulatov. A graph of a relational structure and constarint satisfaction problmes. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004)*. IEEE Computer Society, 2004.

[10] Stanley Burris and H.P. Sankappanavar. *A Course in Universal Algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, 1981.

[11] David Cohen and Peter Jeavons. The complexity of constraint languages. In *Handbook of Constraint Programming*. Elsevier, 2006. Chapter 8.

[12] David Cohen, Peter Jeavons, and Marc Gyssens. A unified theory of structural tractability for constraint satisfaction and spread cut decomposition. In *Proceedings of IJCAI'05*, pages 72–77, 2005.

[13] Béla Csákány. Minimal clones a minicourse. *Algebra Universalis*, 54(1):73–89, 2005.

[14] Victor Dalamu. Generalized majority-minority operations are tractable. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*, pages 438–447.

[15] Víctor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *Constraint Programming '99*, Lecture Notes in Computer Science. Springer Verlag, 1999.

[16] T. Feder and Vardi M.Y. Monotone monadic SNP and constraint satisfaction. In *Proceedings of 25th ACM Symposium on the Theory of Comp uting (STOC)*, pages 612–622, 1993.

[17] E.C. Freuder. A sufficient conditions for backtrack free search. *Journal of the Association for Computing Machinery*, 29(1), 1982. 24-32.

[18] Michael R. Garey and David S. Johnson. *Computers and Intractability.* W.H. Freeman and Company, 1979.

[19] Peter Jeavons, Cohen David, and Marc Gyssens. A structural decomposition for hypergraphs. *Contemporary Mathematics*, 178:161–177, 1994.

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

[21] P.G. Jeavons, D.A. Cohen, and M. Cooper. Constraints consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

[22] P.G. Jeavons and M.C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):633–642, 1995.

[23] Keith A. Kearnes and A Szendrei. The classification of commuative minimal clones. *DISCUSSIONES MATHEMATICAE ALGEBRA AND STOCHASTIC METHODS*, 1999.

[24] A K Mackworth and E C Freuder. The complexity of constraint satisfaction revisited. *Artificial Intelligence*, 59:57–62, 1993.

[25] A.K. Mackworth. *Encyclopedia of Artificial Intelligence*, chapter Constraint Satisfaction, pages 285–293. John Wiley and Sons, 1992.

[26] R.N. McKenzie, G.F. McNulty, and W.F. Taylor. *Algebras, Lattices and Varieties, volume 1.* Wadsworth and Brooks, California, 1987.

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

[28] Reinhard Pöschel. A general Galois theory for operations and relations and concrete charactrerization of related algebraic structures. Technical Report Report R-01/08, Akademie Der Wissenschaftern Der DDR Zentralinstitut für Mathematik und Mechanik, Berlin, 1980.

[29] Combinatorial problems raised from 2-semilattice. Bulatov, andrei. *Journal of Algebra*, to appear.

[30] Robert W. Quakenbush. A survey of minimal clones. *Aequationes Mathematicae*, 50:3–16, 1995.

[31] Jean-Charles Régin. A filtering algorithm for constraints of difference in CSPs. In *Proceedings of AAAI'94*, pages 362–367. AAAI Press, 1994.

[32] I.G. Rosenberg. Minimal clones I: the five types. In *Lectures in Universal Algebra (Proc. Conf. Szeged 1983)*, volume 43 of *Colloq. Math. Soc. Janos Bolyai*, pages 405–427. North-Holland, 1986.

[33] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *Handbook of Constraint Programming*. Elsevier Scince, 2006.

[34] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.

[35] Barbara M. Smith, Sally C. Brailsford, Peter M. Hubbard, and H. Paul Williams. The progressive party problem: Integer linear programming and constraint programming compared. *Constraints*, 1:119–138, 1996.

[36] A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminairs de Mathematiques Superieures*. University of Montreal, 1986.

[37] Pascal Van Hentenryck. *Constraint Satisfaction in Logic Programming*. The MIT Press, 1989.

[38] Pascal Van Hentenryck. Constraint and integer programming in OPL. *INFORMS Journal on Computing*, 14(4):345–372, 2002.