Novi Sad J. Math.
Vol. 40, No. 2, 2010, 95-109

POLYMORPHISMS OF SMALL DIGRAPHS[1]

Libor Barto[2], David Stanovský[3]

Abstract. For each digraph with at most 5 vertices, we provide a list of its polymorphisms interesting with respect to the complexity of the corresponding Constraint Satisfaction Problem. We find a digraph on six vertices such that the complexity of its retraction problem is unknown with current techniques; this is the smallest such example.

AMS Mathematics Subject Classification (2000): 05C85, 68R10, 08A70

Key words and phrases: Polymorphisms of digraphs, constraint satisfaction problem

1. Introduction

For a fixed finite relational structure $\mathbb{A}$, the Constraint Satisfaction Problem over $\mathbb{A}$, CSP($\mathbb{A}$) for short, can be formulated as a decision problem asking whether an input (finite) relational structure $\mathbb{X}$ of the same type as $\mathbb{A}$ admits a homomorphism to $\mathbb{A}$. The computational and descriptive complexity of CSPs is now a very active research area, the main open question is the Dichotomy Conjecture of Feder and Vardi [18] stating that each CSP($\mathbb{A}$) is either NP-complete, or tractable. (It is assumed $P \neq NP$ throughout the paper.)

Much of recent development builds on results in [21, 11, 12], where it was shown that the complexity of CSP($\mathbb{A}$) depends only on a certain set of functions – the polymorphisms of the structure $\mathbb{A}$. Bulatov, Jeavons and Krokhin [11] formulated the Algebraic Dichotomy Conjecture postulating a necessary and sufficient condition for tractability of CSP($\mathbb{A}$) in terms of polymorphisms (and they proved NP-completeness when the condition is not met). The conjecture was confirmed in various special cases, the most notable recent results are [6, 7, 4, 1].

This paper gives a report on results obtained by computer testing the existence of various types of polymorphisms for digraphs, i.e. relational structures with one binary relation. We have chosen those polymorphisms which are important in the CSP and which can be checked in reasonable time – Mal'cev polymorphism, near-unanimity polymorphisms (up to arity 5), edge polymorphisms (up to arity 5), semilattice polymorphism, 2-semilattice polymorphism,

[1] This work is a part of the research project MSM 0021620839 financed by MŠMT ČR, and was encountered within the framework of the Czech-Hungarian cooperation MEB 040915. The authors were also partly supported by their GAČR grants, 201/09/P223, and 201/08/P056, respectively.

[2] Charles University in Prague, Faculty of Mathematics and Physics, KA MFF UK, Sokolovská 83, 18675 Praha 8, Czech Republic, email: barto@karlin.mff.cuni.cz

[3] Charles University in Prague, Faculty of Mathematics and Physics, KA MFF UK, Sokolovská 83, 18675 Praha 8, Czech Republic, email: stanovsk@karlin.mff.cuni.cz

totally symmetric polymorphisms (up to arity 4), weak near-unanimity polymorphisms (up to arity 5), polymorphisms characterizing bounded width, and Siggers polymorphism (see Section 2 for definitions and explanation).

We concentrated on digraphs for two reasons. First, they are simple enough so that they are ideal toy examples for testing various conjectures (not only in CSP). Secondly, they are general enough for studying the complexity of CSPs – a construction in [18] shows that for every relational structure $\mathbb{A}$ there exists a digraph $\mathbb{G}$ such that CSP($\mathbb{A}$) and CSP($\mathbb{G}$) are poly-time equivalent.

The above mentioned polymorphisms were checked for all non-isomorphic digraphs with up to 5 vertices and for a number of randomly chosen digraphs on 6, 7 and 8 vertices. The results are discussed in Section 3 and our computational techniques in Section 4. We believe that our rather comprehensive catalogue of polymorphisms will be useful for researchers in CSP and universal algebra. Moreover, we have found a digraph on 6 vertices, which is conjecturally tractable according to the Algebraic Dichotomy Conjecture, but the complexity of the corresponding CSP (more precisely, the complexity of its retraction problem, see Section 2) is unknown with current techniques. The digraph is depicted on Figure 1. Note that this is the smallest such example, since we found that all digraphs with at most 5 vertices are either NP-complete, or have bounded width (and hence are tractable).

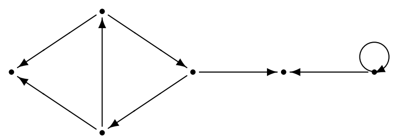

Figure 1: The *sea devil* digraph: the smallest digraph with (currently) unknown complexity of its CSP.

2. CSP and Polymorphisms

A *digraph* is a pair $\mathbb{G} = (V, E)$, where V is a finite set of vertices and $E \subseteq V \times V$ is a set of edges. For a digraph $\mathbb{G} = (V = \{v_1, \dots, v_k\}, E)$ we denote by $\bar{\mathbb{G}}$ the relational structure $\bar{\mathbb{G}} = (V, E, \{v_1\}, \{v_2\}, \dots, \{v_k\})$, so that $\bar{\mathbb{G}}$ has one binary relation E and $|V|$ (singleton) unary relations. The CSP over $\mathbb{G}$ is also called $\mathbb{G}$-coloring problem, CSP($\bar{\mathbb{G}}$) is the retraction problem for $\mathbb{G}$.

A digraph is a core if all of its endomorphisms are bijective. It is easily seen that every digraph $\mathbb{G}$ has an induced core subgraph $\mathbb{H}$ (unique up to isomorphism) such that $\mathbb{G}$ maps homomorphically onto $\mathbb{H}$ and therefore the corresponding CSPs are the same. It was shown in [11] that for a core digraph $\mathbb{G}$ the problems CSP($\mathbb{G}$) and CSP($\bar{\mathbb{G}}$) are poly-time equivalent. Therefore the results below which are formulated for $\bar{\mathbb{G}}$ can be translated to results about $\mathbb{G}$.

An n-ary *polymorphism* of a digraph $\mathbb{G} = (V, E)$ is a mapping $f : V^n \to V$

which preserves edges. Namely, for any $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n) \in E$, the pair $(f(a_1, \dots, a_n), f(b_1, \dots, b_n))$ is in E.

A polymorphism f is called *idempotent*, if $f(x, x, \dots, x) = x$ for all $x \in V$. The following definition lists some important types of idempotent polymorphisms.

Definition 1. Let $n \geq 2$. An idempotent polymorphism $f : V^n \to V$ is

- a *weak near-unanimity polymorphism* (wnu n), if (for all $x, y \in V$)

$$f(y, x, x, \dots, x) = f(x, y, x, x, \dots, x) = \dots = f(x, x, \dots, x, y);$$

- a *near-unanimity polymorphism* (nu n), if it is a weak near-unanimity polymorphism of arity $n \geq 3$ and $f(x, x, \dots, x, y) = x$;

- a *totally symmetric polymorphism* (ts n), if

$$f(x_1, x_2, \dots, x_n) = f(y_1, y_2, \dots, y_n)$$

whenever $\{x_1, \dots, x_n\} = \{y_1, \dots, y_n\}$;

- an *edge polymorphism* (edge n), if $n \geq 3$,

$$f(y, y, x, x, \dots, x) = t(y, x, y, x, x, \dots, x) = x$$

and

$$f(x, x, x, y, x, \dots, x) = f(x, x, x, x, y, x, \dots, x) = \dots = f(x, \dots, x, y) = x;$$

- a *Siggers polymorphism* (siggers), if $n = 4$ and

$$f(x, y, y, z) = f(y, x, z, x);$$

- a *Mal'cev polymorphism* (malcev), if $n = 3$ and

$$f(y, y, x) = f(x, y, y) = x;$$

- a *semilattice polymorphism* (sml), if $n = 2$ and

$$f(x, y) = f(y, x), \quad f(f(x, y), z) = f(x, f(y, z));$$

- a 2-*semilattice polymorphism* (2sml), if $n = 2$ and

$$f(x, y) = f(y, x), \quad f(f(x, y), x) = f(x, y);$$

The poset on Figure 2 compares the strength of these polymorphisms. An edge means that the lower polymorphism implies the upper one. In other words, the lower condition on polymorphisms is stronger than the upper one. Most of the implications, which are not explained below, are readily seen from the fact

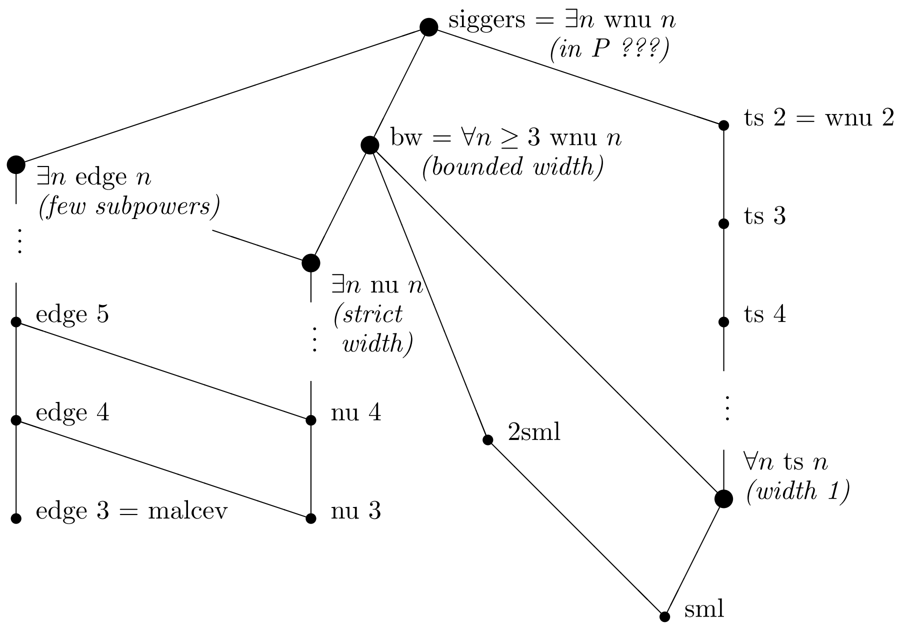

Figure 2: The poset of polymorphism conditions: theoretical picture.

that the set of polymorphisms is closed under composition. To avoid confusion, we remark that a k-ary edge polymorphism is usually called a $(k-1)$-edge polymorphism in the literature.

Bulatov, Jeavons and Krokhin [11] proved that the complexity of CSP($\bar{\mathbb{G}}$) depends only on the set of idempotent polymorphisms of $\mathbb{G}$. Moreover, they showed that the absence of so called Taylor polymorphism (not defined above) implies NP-completness of CSP($\bar{\mathbb{G}}$) and they conjectured that if $\bar{\mathbb{G}}$ admits such a polymorphism, then CSP($\bar{\mathbb{G}}$) is tractable. This is the Algebraic Dichotomy Conjecture mentioned in the introduction. We remark that the formulation in [11] was different, but equivalent according to results in Chapter 9 of [19] (see also [14]). The existence of a Taylor polymorphism was shown to be equivalent to the existence of a weak near-unanimity polymorphism [26], and, more recently, to the existence of a cyclic polymorphism [3]. For computational purposes we used yet another equivalent characterization, the existence of what we call here a Siggers polymorphism. Note that the original condition of Siggers [27] uses 6-ary polymorphism. It was observed by P. Marković and R. McKenzie that from a result of [4] one can derive many similar conditions (see [27]); we chose one that seemed computationally most feasible. We remark that this condition is the weakest nontrivial "equational condition" on idempotent polymorphisms – all strictly weaker conditions are satisfied by all digraphs (see Chapter 9 of [19] for a more general result).

There are two known algorithms, or, rather, algorithmic principles for solving CSP($\bar{\mathbb{G}}$) in poly-time. All of known algorithms are combinations of these two.

The first algorithmic idea tries to describe all solutions (=homomorphisms) in a way similar to Gaussian elimination. The milestone result [10] was an algorithm that works for all digraphs (more generally, all relational structures) with a Mal'cev polymorphism. The result was generalized in [5, 20] to structures with an edge polymorphism (note that a ternary edge polymorphism is basically Mal'cev after shifting variables). The last result in certain sense (see [20]) finishes the research in this direction. The algorithm solving CSP($\bar{\mathbb{G}}$) in this case is called *few subpowers*. If $\mathbb{G}$ admits an edge term we also say that $\mathbb{G}$ has few subpowers. It was recently shown [22] that digraphs with Mal'cev polymorphism have a very restrictive structure, in particular, they all have a near-unanimity polymorphism of arity 3 (hence, considering digraphs only, the malcev node in Figure 2 could be shifted below nu3).

The second algorithmic idea is to try to check whether a homomorphism exists by looking at small pieces of the input digraph. The digraphs for which such a local consistency algorithm can be used are said to have *bounded width* (bw), see [18, 24, 13]. It was shown that the existence of a near-unanimity polymorphism implies bounded width [18]; near unanimity polymorphisms actually characterize a stronger condition than bounded width, called *strict width* (see [18]). A semilattice polymorphism implies bounded width as well [21]. More generally, totally symmetric polymorphisms of all arities ensure bounded width, they characterize problems of *width* 1 [17]. The semilattice case was generalized by Bulatov [8] – he proved that a 2-semilattice polymorphism suffices. The research in this direction was finished after a sequence of partial results [23, 15, 2] in [1], where the authors proved the Bounded Width Conjecture of Larose and Zádori from [24] – a digraph has bounded width if and only if it has weak near-unanimity polymorphisms of all but finitely many arities (an independent proof was announced by Bulatov [9]). Using results of [1] it was shown by M. Kozik (a manuscript is not yet available) that this condition is equivalent to the existence of weak near-unanimity polymorphisms f of arity 3 and g of arity 4 such that $f(x, x, y) = g(x, x, x, y)$. This characterization was used in our computations (the condition (bw)).

3. Results

In our experiments, we had to restrict the arities of polymorphisms. We searched for (weak) near-unanimity and edge polymorphisms up to arity 5 and for totally symmetric polymorphisms up to arity 4. The corresponding poset is drawn on Figure 3.

The results and scripts running the computation are available from our website

`http://www.karlin.mff.cuni.cz/~stanovsk/math/gpoly.htm`

In this section we present several interesting statistics.

3.1. Digraphs on ≤ 5 vertices

The major part of our computation was, to determine polymorphisms of all digraphs (all up to isomorphism) with at most 5 vertices.

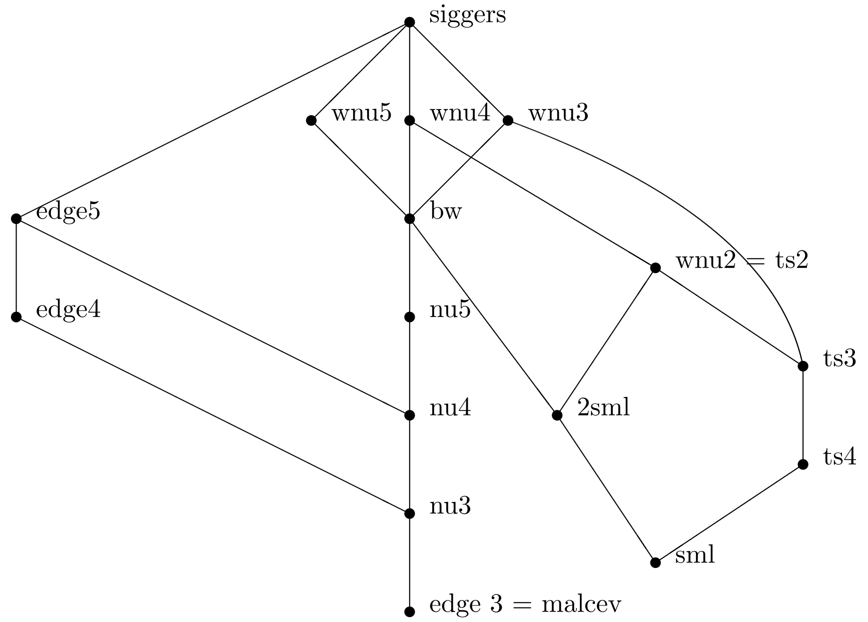

Figure 3: The poset of polymorphism conditions: our setting.

On Figures 4, 5, 6 and 7, one can read dependencies between the polymorphism conditions for digraphs with 2, 3, 4, and 5 vertices, respectively. The entries in the table show the number of digraphs that have both the row and the column polymorphisms. Particularly, the diagonal shows the total number of digraphs with the respective polymorphism.

	siggers	edge5	edge4	nu5	nu4	nu3	malcev	2sml	sml	wnu2	ts34
siggers	**10**	10	10	10	10	10	8	9	9	9	9
edge5	10	**10**	10	10	10	10	8	9	9	9	9
edge4	10	10	**10**	10	10	10	8	9	9	9	9
nu5	10	10	10	**10**	10	10	8	9	9	9	9
nu4	10	10	10	10	**10**	10	8	9	9	9	9
nu3	10	10	10	10	10	**10**	8	9	9	9	9
malcev	8	8	8	8	8	8	**8**	7	7	7	7
2sml	9	9	9	9	9	9	7	**9**	9	9	9
sml	9	9	9	9	9	9	7	9	**9**	9	9
wnu2	9	9	9	9	9	9	7	9	9	**9**	9
ts34	9	9	9	9	9	9	7	9	9	9	**9**

Figure 4: Dependence table for 2-element digraphs.

Figures 8, 9 and 10 display the poset of polymorphism conditions for digraphs with 2, 3, and 4/5 vertices, respectively (for sizes 4 and 5, the poset is identical). For small digraphs, some of the conditions are equivalent; so, what you see, are factor posets of the poset from Figure 3 over this equivalence. Vertices with multiple labels represent conditions that are indistinguishable on digraphs of

	siggers	edge5	edge4	nu5	nu4	nu3	malcev	2sml	sml	wnu2	ts34
siggers	**97**	92	92	92	92	92	29	91	90	91	90
edge5	92	**92**	92	92	92	92	29	86	85	86	85
edge4	92	92	**92**	92	92	92	29	86	85	86	85
nu5	92	92	92	**92**	92	92	29	86	85	86	85
nu4	92	92	92	92	**92**	92	29	86	85	86	85
nu3	92	92	92	92	92	**92**	29	86	85	86	85
malcev	29	29	29	29	29	29	**29**	25	24	25	24
2sml	91	86	86	86	86	86	25	**91**	90	91	90
sml	90	85	85	85	85	85	24	90	**90**	90	90
wnu2	91	86	86	86	86	86	25	91	90	**91**	90
ts34	90	85	85	85	85	85	24	90	90	90	**90**

Figure 5: Dependence table for 3-element digraphs.

	siggers	edge5	edge4	nu5	nu4	nu3	malcev	2sml	sml	wnu2	ts34
siggers	**2279**	1736	1690	1737	1736	1690	118	2190	2178	2190	2184
edge5	1736	**1736**	1690	1736	1736	1690	118	1676	1670	1676	1670
edge4	1690	1690	**1690**	1690	1690	1690	118	1630	1624	1630	1624
nu5	1737	1736	1690	**1737**	1736	1690	118	1677	1671	1677	1671
nu4	1736	1736	1690	1736	**1736**	1690	118	1676	1670	1676	1670
nu3	1690	1690	1690	1690	1690	**1690**	118	1630	1624	1630	1624
malcev	118	118	118	118	118	118	**118**	96	92	96	92
2sml	2190	1676	1630	1677	1676	1630	96	**2190**	2178	2190	2184
sml	2178	1670	1624	1671	1670	1624	92	2178	**2178**	2178	2178
wnu2	2190	1676	1630	1677	1676	1630	96	2190	2178	**2190**	2184
ts34	2184	1670	1624	1671	1670	1624	92	2184	2178	2184	**2184**

Figure 6: Dependence table for 4-element digraphs.

	siggers	edge5	edge4	nu5	nu4	nu3	malcev	2sml	sml	wnu2	ts34
siggers	**136819**	69444	60528	70832	69444	60528	472	132510	130871	132510	132430
edge5	69444	**69444**	60528	69444	69444	60528	472	68610	68554	68610	68555
edge4	60528	60528	**60528**	60528	60528	60528	472	59694	59640	59694	59641
nu5	70832	69444	60528	**70832**	69444	60528	472	69998	69940	69998	69943
nu4	69444	69444	60528	69444	**69444**	60528	472	68610	68554	68610	68555
nu3	60528	60528	60528	60528	60528	**60528**	472	59694	59640	59694	59641
malcev	472	472	472	472	472	472	**472**	383	359	383	360
2sml	132510	68610	59694	69998	68610	59694	383	**132510**	130871	132510	132430
sml	130871	68554	59640	69940	68554	59640	359	130871	**130871**	130871	130871
wnu2	132510	68610	59694	69998	68610	59694	383	132510	130871	**132510**	132430
ts34	132430	68555	59641	69943	68555	59641	360	132430	130871	132430	**132430**

Figure 7: Dependence table for 5-element digraphs.

respective size. These posets can be read from Figures 4, 5, 6 and 7. Indeed, the implication "polymorphism A implies polymorphism B for all digraphs of a given size" holds precisely if the number of digraphs of the given size admitting both A, B is the same as the number of digraphs admitting A.

Figure 11 displays the number of digraphs on 2 to 5 vertices, where a given polymorphism condition is *minimal*, that is, where the digraph has the given polymorphism, but does not admit any stronger one with respect to Figure 3. It turns out that many digraphs admit a rather low condition: the most popular one is semilattice, and many digraphs also admit ternary nu. We note that Mal'cev polymorphisms are rare, a fact that is theoretically explained in [22].

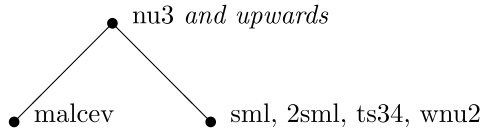

Figure 8: Poset of polymorphism conditions on 2-element digraphs.

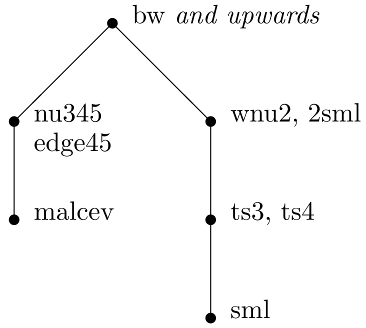

Figure 9: Poset of polymorphism conditions on 3-element digraphs.

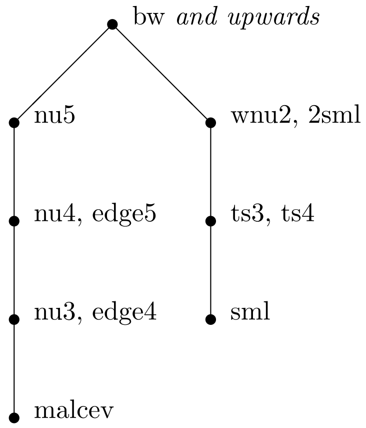

Figure 10: Poset of polymorphism conditions on 4- and 5-element digraphs.

The first line of the table shows the number of NP-complete digraphs, the last line the total number of non-isomorphic digraphs.

Figure 12 displays the smallest digraphs such that a given condition is minimal. In cases when there are more examples of the same size, the lexicographically first one is chosen. Digraphs are presented both by their size/id number, and row vectors of their incidence matrices; examples with 6 vertices were found by random search (see below). Included are also digraphs that provide independence of the two branches from Figure 9 (a digraph that admits Mal'cev polymorphism and avoids binary weak near-unanimity polymorphism, and a digraph that admits semilattice polymorphism and avoids 5-ary near-unanimity polymorphism), and digraphs that provide independence of the two branches of the pentagon on Figure 3.

The number of digraphs with a given set of minimal polymorphism conditions is summarized on Figure 13. Particularly interesting are the three bottom lines of the table – certain combinations of conditions are satisfied by precisely 1 or 2 digraphs (out of 291968) on 5 vertices.

Figures 14 and 15 contain the minimal polymorphism conditions for all digraphs on 2 and 3 vertices. The full list for 4- and 5-element digraphs can be downloaded from our website (the size 5 tables have over 1000 pages!).

condition	size 2	size 3	size 4	size 5
NONE	0	7	765	155151
malcev	8	29	118	471
nu3	2	63	1572	60056
nu4	0	0	46	8916
nu5	0	0	1	1388
edge4	0	0	0	0
edge5	0	0	0	0
bw	0	0	29	3475
sml	9	90	2178	130870
2sml	0	1	12	1639
ts4	0	0	6	1559
ts3	0	0	0	0
wnu2	0	0	0	0
wnu3	0	0	0	0
wnu4	0	0	0	0
wnu5	0	0	0	0
siggers	0	0	0	0
TOTAL	10	104	3044	291968

Figure 11: The number of digraphs where a given polymorphism condition is minimal.

3.2. Larger digraphs

There are 96,928,992 non-isomorphic digraphs on 6 vertices. This makes the brute-force computation through all of them nearly impossible. Instead, we were generating random digraphs with given edge probabilities. We experimented with ratios from 1/6 to 5/6, using weeks of computer time. We obtained two interesting digraphs, presented in Figure 12.

Particularly, we found the *sea devil* digraph, the smallest digraph with a Siggers polymorphism that does not have bounded width (see Figure 1). It has a 3-ary and 5-ary wnu, but fails to have a 4-ary one. Separately, we computed that it fails to have edge polymorphisms up to arity 7. It is unknown to us,

condition	size/#	example
nu3	2/7	01 11
nu4	4/948	0111 0011 0000 1101
nu5	4/2048	0111 1010 1011 1111
bw	4/220	0011 0000 1001 0100
2sml	3/21	010 001 100
ts4	4/1097	0111 0010 0100 1001
wnu3,wnu5	6/random	000000 000000 010010 010100 100001 101000
has sml, no nu5	3/71	011 010 101
has malcev, no wnu2	3/4	001 000 100
has 2sml, no ts3	3/21	010 001 100
has ts4, no 2sml	6/random	011000 000100 000001 010000 010101 000001

Figure 12: The smallest digraphs where a given polymorphism condition is minimal, and the smallest independence examples.

minimal poly's	size 2	size 3	size 4	size 5	example
malcev sml	7	24	92	358	00 00
nu3 sml	2	61	1532	59281	01 11
malcev	1	4	22	89	01 10
sml	0	5	507	60931	011 010 101
nu3	0	2	38	745	011 001 010
malcev 2sml	0	1	4	23	010 001 100
nu4 sml	0	0	46	8914	0111 0011 0000 1101
bw	0	0	29	3475	0011 0000 1001 0100
2sml ts4	0	0	6	1556	0111 0010 0100 1001
nu5 sml	0	0	1	1386	0111 1010 1011 1111
nu3 2sml	0	0	2	30	0111 0010 0001 0100
2sml	0	0	0	25	00101 00000 00011 10001 01000
nu5 2sml ts4	0	0	0	2	00011 00100 10011 11000 11001
nu4 2sml	0	0	0	2	00111 00011 00010 00001 00100
malcev 2sml ts4	0	0	0	1	00100 00010 00001 10000 01000
NONE	0	7	765	155151	011 001 100

Figure 13: The number of digraphs with given combination of minimal polymorphism conditions.

whether the digraph has an edge polymorphism of a higher arity, in other words, whether its retraction problem can be solved by the few subpowers algorithm.

The sea devil digraph was found using edge probability 1/3. It is essentially the only such a 6-element digraph we found (one can obtain few more by reversing certain edges.) We also found three such digraphs on 7 vertices using probabilities 1/3 and 1/4 and one on 8 vertices (1/4), but all of them are just simple modifications of the sea devil.

#	table	minimal polymorphisms
01	00 00	malcev sml
02	01 00	malcev sml
03	01 10	malcev
04	00 01	malcev sml
05	01 01	malcev sml
06	00 11	malcev sml
07	01 11	nu3 sml
08	10 01	malcev sml
09	11 01	nu3 sml
10	11 11	malcev sml

Figure 14: Minimal polymorphism conditions on 2-element digraphs.

The digraph proving independence of 4-ary totally symmetric and 2-semilattice polymorphisms is tractable: it has bounded width. We obtained two digraphs with the same properties, i.e., with minimal conditions ts4 and bw: the one presented in Figure 12 (edge probability 1/3), and a very different example on 7 vertices (1/2), with incidence matrix

0010000 0001100 0011000 0111100 1101010 0110101 1011010.

4. Techniques

In the present section, we give a short account on how we conducted the search. Shortly speaking, we generated a list of all digraphs of a given size, and for each of them were running tests for polymorphism conditions, starting from the bottom of the poset on Figure 3. This way, we obtained all minimal polymorphism conditions. For sizes more than 5, there are too many digraphs. Instead of generating them all, we were taking random digraphs, with various edge probabilities.

To generate all digraphs of given size up to isomorphism, we used a Java applet by Miklós Maróti [25]. It can generate all 291,968 non-isomorphic digraphs on 5 vertices within few minutes.

We created a Perl script that reads through a list of digraphs, and for each of them determines given polymorphism conditions. To check whether a given digraph has a given polymorphism, we use the finite model builder Paradox [16]. This is a program that reads a set of first order formulas, and searches for a (finite) model. It translates the problem into SAT, calls an external SAT-solver (miniSAT, in this case), and interprets the result. To set up an input file for Paradox, we encode a digraph by its incidence table, add axioms to ensure the universe of the model is the set of vertices, put the equations we impose on the polymorphism we search, and add the conditions that all mappings involved preserve the edges. A sample file is displayed on Figure 16. When Paradox

#	table	minimal poly's
01	000 000 000	malcev sml
02	001 000 000	malcev sml
03	001 001 000	malcev sml
04	001 000 100	malcev
05	000 001 100	malcev sml
06	001 001 100	malcev
07	000 000 110	malcev sml
08	001 000 110	malcev
09	001 001 110	malcev
10	000 000 001	malcev sml
11	001 000 001	malcev sml
12	001 001 001	malcev sml
13	000 000 101	malcev sml
14	001 000 101	nu3 sml
15	000 001 101	nu3 sml
16	001 001 101	nu3 sml
17	000 000 111	malcev sml
18	001 000 111	nu3 sml
19	001 001 111	nu3 sml
20	011 001 000	nu3 sml
21	010 001 100	malcev 2sml
22	011 001 100	
23	011 001 010	nu3
24	011 000 110	nu3
25	011 001 110	
26	010 000 001	malcev sml
27	011 000 001	nu3 sml
28	010 001 001	malcev sml
29	011 001 001	nu3 sml
30	010 000 101	malcev sml
31	011 000 101	nu3 sml
32	010 001 101	nu3 sml
33	011 001 101	nu3 sml
34	010 000 011	nu3 sml
35	011 000 011	malcev sml
36	010 001 011	nu3 sml
37	011 001 011	nu3 sml
38	010 000 111	nu3 sml
39	011 000 111	nu3 sml
40	010 001 111	nu3 sml
41	011 001 111	nu3 sml
42	011 101 110	
43	010 100 001	malcev sml
44	011 100 001	nu3 sml
45	011 101 001	nu3 sml
46	010 100 101	nu3 sml
47	011 100 101	nu3 sml
48	010 101 101	nu3 sml
49	011 101 101	nu3 sml
50	010 100 111	nu3 sml
51	011 100 111	nu3 sml
52	011 101 111	nu3 sml

#	table	minimal poly's
53	000 010 001	malcev sml
54	001 010 001	malcev sml
55	000 011 001	nu3 sml
56	001 011 001	nu3 sml
57	000 010 101	malcev sml
58	001 010 101	nu3 sml
59	000 011 101	nu3 sml
60	001 011 101	nu3 sml
61	001 010 011	nu3 sml
62	000 011 011	malcev sml
63	001 011 011	nu3 sml
64	000 010 111	nu3 sml
65	001 010 111	nu3 sml
66	000 011 111	nu3 sml
67	001 011 111	nu3 sml
68	011 010 001	nu3 sml
69	011 011 001	nu3 sml
70	010 010 101	nu3 sml
71	011 010 101	sml
72	010 011 101	
73	011 011 101	sml
74	011 011 011	malcev sml
75	010 010 111	nu3 sml
76	011 010 111	nu3 sml
77	010 011 111	nu3 sml
78	011 011 111	nu3 sml
79	000 110 101	nu3 sml
80	001 110 101	sml
81	000 111 101	nu3 sml
82	001 111 101	nu3 sml
83	001 110 111	sml
84	000 111 111	malcev sml
85	001 111 111	nu3 sml
86	011 110 101	
87	011 111 101	
88	011 111 111	nu3 sml
89	100 010 001	malcev sml
90	101 010 001	nu3 sml
91	101 011 001	nu3 sml
92	101 010 101	malcev sml
93	100 011 101	nu3 sml
94	101 011 101	nu3 sml
95	100 010 111	nu3 sml
96	101 010 111	nu3 sml
97	101 011 111	nu3 sml
98	111 011 001	nu3 sml
99	110 011 101	
100	111 011 101	sml
101	111 011 011	nu3 sml
102	111 010 111	nu3 sml
103	111 011 111	nu3 sml
104	111 111 111	malcev sml

Figure 15: Minimal polymorphism conditions on 3-element digraphs.

finishes its job, the script reads the answer and, in case of failure, it proceeds further up in the polymorphism poset.

Running the poset bottom up (rather than top down) makes better sense for small digraphs, since many of them actually satisfy very low polymorphism conditions (see Figure 11). Also, bottom up direction is more likely to avoid costly computation of polymorphisms of high arities. For larger sizes, however,

```
cnf(sml,axiom, t(X,X)=X ).
cnf(sml,axiom, t(X,Y)=t(Y,X) ).
cnf(sml,axiom, t(X,t(Y,Z))=t(t(X,Y),Z) ).
cnf(pr,axiom,(~gr(X0,X1)|~gr(X2,X3)|gr(t(X0,X2),t(X1,X3)) )).
cnf(graph,axiom,~gr(n0,n0)).
cnf(graph,axiom,gr(n0,n1)).
cnf(graph,axiom,gr(n1,n0)).
cnf(graph,axiom,gr(n1,n1)).
cnf(elems,axiom,n0!=n1).
cnf(elems,axiom,(X=n0|X=n1)).
```

Figure 16: Paradox input file asking to determine whether the 2-element digraph #07 admits a semilattice polymorphism.

it is worth starting with checking the Siggers polymorphism, since most digraphs are NP-complete.

We imposed time limit of 60 seconds for each call of Paradox. The time limit was never met for digraphs of size ≤ 5. The following table shows computation times for a single Paradox call.

- sizes 2,3: always fractions of seconds,
- size 4: always below 2 seconds,
- size 5:
  - 2.9 million (83%) in < 1 second,
  - 0.6 million (17%) between 1 and 2 seconds,
  - 505 between 2 and 3 seconds,
  - 91 between 3 and 4 seconds, etc.
  - 8 computations over 10 seconds, with maximum 21s to compute that the digraph #235816 admits nu5.

The total time to process all 5-element digraphs was a little over a week of computer time.

For larger digraphs, we used a pseudorandom generator. We modified the script, to generate digraphs with given edge probability (instead of reading through a file). Timeouts were met quite often, particularly for the 5-ary operations. It turned out that in most of the difficult tasks, the polymorphism actually existed. It means that proving nonexistence seems to be an easier problem than finding an example.

Finally, let us note that, in the initial experiments, we tried two model builders, Paradox and Mace4 (both based on SAT). We realized soon that Paradox outperforms Mace4 by an order of magnitude. The model builders based on different methods (iProver, Geo) do not seem to handle this sort of tasks efficiently.

References

[1] Barto, L., Kozik, M., Constraint Satisfaction Problems of Bounded Width. In: Proceedings of IEEE Symposium on Foundations of Computer Science, 2009, 595–603.

[2] Barto, L., Kozik, M., Congruence Distributivity Implies Bounded Width. SIAM J. Comput., 39 (2009), 1531–1542.

[3] Barto, L., Kozik, M., Absorbing subalgebras, cyclic terms and the constraint satisfaction problem. Submitted.

[4] Barto, L., Kozik, M., Niven, T., The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). SIAM J. Comput., 38 (2008), 1782–1802.

[5] Berman, J., Idziak, P., Marković, P., McKenzie, R., Valeriote, M., Willard, R., Varieties with few subalgebras of powers. Trans. Amer. Math. Soc., 362/3 (2010), 1445-1473.

[6] Bulatov, A., Tractable conservative Constraint Satisfaction Problems. In: Proceedings of Symposium on Logic in Computer Science, 2003.

[7] Bulatov, A., A dichotomy theorem for constraint satisfaction problems on a 3-element set. J. ACM, 53/1 (2006), 66-120.

[8] Bulatov, A., Combinatorial problems raised from 2-semilattices. J. Algebra, 298 (2006), 321–339.

[9] Bulatov, A., Bounded relational width. Manuscript, 2010.

[10] Bulatov, A., Dalmau, V., A simple algorithm for Mal'tsev constraints. SIAM J. Comput. 36/1 (2006), 16–27.

[11] Bulatov, A., Krokhin, A., Jeavons, P., Constraint satisfaction problems and finite algebras. In: Lecture Notes in Comput. Sci. 1853 (2000), 272–282.

[12] Bulatov, A., Krokhin, A., Jeavons, P., Classifying the complexity of constraints using finite algebras. SIAM J. Comput., 34/3 (2005), 720–742.

[13] Bulatov, A., Krokhin, A., Larose, B., Dualities for Constraint Satisfaction Problems. In: Complexity of Constraints: An Overview of Current Research Themes (2008), 93–124.

[14] Bulatov, A., Valeriote, M., Recent Results on the Algebraic Approach to the CSP. In: Lecture Notes in Computer Science 5250 (2008), 68–92.

[15] Carvalho, C., Dalmau, V., Marković, P., Maróti, M., CD(4) has bounded width. Algebra Universalis, 60/3 (2009), 293–307.

[16] Claessen, K., Sorensson, N., New techniques that improve MACE-style finite model finding. In: Proceedings of the CADE-19 Workshop: Model Computation - Principles, Algorithms, Applications (2003).

[17] Dalmau, V., Pearson, J., Closure Functions and Width 1 Problems. In: CP 1999, 159–173.

[18] Feder, T., Vardi, M., The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. SIAM J. Comput., 28 (1999), 57–104.

[19] Hobby, D., McKenzie, R., The structure of finite algebras. Providence, RI: American Mathematical Society 1988.

[20] Idziak P., Marković, P., McKenzie, R., Valeriote, M., Willard, R., Tractability and learnability arising from algebras with few subpowers. In: Proceedings of IEEE Symposium on Logic in Computer Science, 2007, 213–222.

[21] Jeavons, P., Cohen, D., Gyssens, M., Closure properties of constraints. J. ACM 44/4 (1997), 527–548.

[22] Kazda, A., Maltsev digraphs have a majority polymorphism. European Journal of Combinatorics, 32 (2011), 390–397.

[23] Kiss, E., Valeriote, M., On tractability and congruence distributivity. Log. Methods Comput. Sci., 3/2 (2007).

[24] Larose, B., Zádori, L., Bounded width problems and algebras. Algebra Universalis, 56/3-4 (2007), 439–466.

[25] Maróti, M., `http://www.math.u-szeged.hu/~mmaroti/applets/ModelPrinter.html`

[26] Maróti, M., McKenzie, R., Existence theorems for weakly symmetric operations. Algebra Universalis, 59/3-4 (2008), 463–489.

[27] Siggers, M., A strong Mal'cev condition for locally finite varieties omitting the unary type. To appear in Algebra Universalis.

Received by the editors December 21, 2010