Computational General Algebra on Ten Dollars a Day

Peter Jipsen

Chapman University, Orange, California

GAIA 2013, July 15, Melbourne

Once upon a time...

long, long ago...

even before the World Wide Web existed...

specifically in the summer of 1991...

at a NATO sponsored event...

Brian Davey...

gave an excellent series of talks with the title...

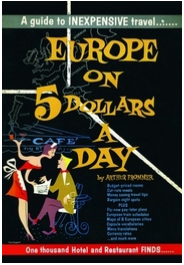
"Duality theory on ten dollars a day"

A very nice paper based on Brian's talks appeared in the proceedings **Algebras and Orders** (ed. I.G. Rosenberg, G. Sabidussi) of the summer school in 1993

The ebook can be **downloaded for free** from Springer

From the abstract: "...The presentation is in the style of a travel guide"

Hence the title, from the classic "Europe on 5 dollars a day"



1957, updated to "$10" in 1976... "$85" in 2004

The title of the current talk is, however, meant more literally:

How much computation is possible if one spends $10 per day on electricity?

Computational Science: using large scale **computation** to support **theoretical science** and **experimental science** by simulating systems, testing models and analyzing big data sets

E.g. computational biology, computational chemistry, computational physics

and **computational mathematics**: applied mathematics, operations research

but also **computational group theory** (e.g. GAP, Magma)
computational geometry (e.g. Flyspeck)
computational ring theory (e.g. Singular, Macauley)
computational number theory (e.g. GIMPS)

From Wikipedia: A brief history of supercomputing

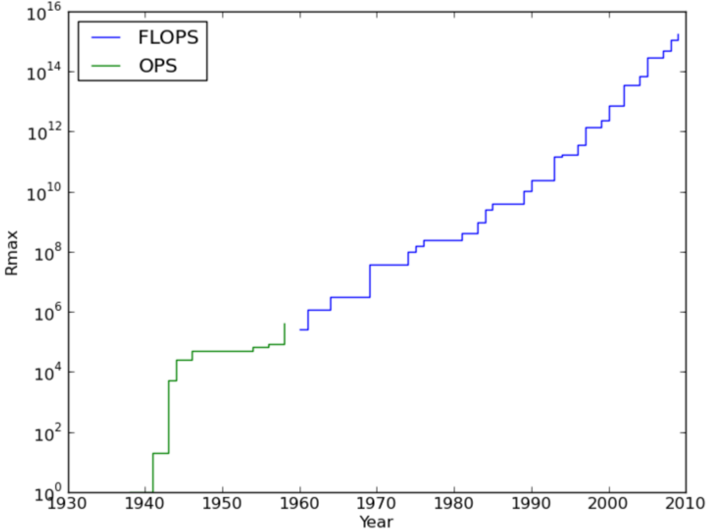
First supercomputer 1964: **CDC 6600** by **Control Data Corporation** designed by **Seymour Cray**

Speed measured in FLOPS = floating point operations per second

Year	Computer	FLOPS
1964	CDC 6600	10^6
1976	Cray 1	10^8
1985	Cray 2	$2 \cdot 10^9$
2008	IBM Roadrunner	10^{15}
2012	Cray Titan	$17 \cdot 10^{15}$
2013	NUDT Tianhe-2	$34 \cdot 10^{15}$

Average recent laptop $\approx 10^{10}$ FLOPS/processor core

Logscale plot of computing speed



Speed increased from 10^6 to $3 \cdot 10^{16}$ in 49 years, so increased by a factor of

$3 \cdot 10^{10} = 2^{34.8}$

$49 * 12/34.8 = 16.9$ months doubling time

Conclusion: Computing power has doubled roughly every 18 months for the last 50 years

Computational universal algebra is not yet making significant use of this exponential growth

Cost of computing for 10^9 FLOPS

1985: \$30 million (Cray XM/P)
1997: \$40000 (Pentium Pro Beowulf clusters)
2003: \$100 (KASY0)
2012: \$0.75 (quad AMD 7970) $4 \cdot 10^{12}$ FLOPS for \$3000

Energy cost for running a supercomputer:

2010: Chinese Tianhe-1A running at $2.5 \cdot 10^{15}$ FLOPS uses 4 MWatts

$\approx$ **\$400/hour** $\approx$ \$10000/day $\approx$ \$3.5 million/year

Efficiency: $6 \cdot 10^8$ FLOPS/Watt

2011: IBM Blue Gene **efficiency** $2 \cdot 10^9$ FLOPS/Watt

How many FLOPS for ten dollars?

1 kWh costs about \$0.10, so \$10 = 100 kWh $\approx$ 4kWday = 4000W all day

= boiling water in two tea kettles (all day long)

$\approx$ running 50 desktop computers, or 150 laptop computers

$\approx 2 \cdot 10^{12}$ FLOPS (for **every second**, all day long)

or $8 \cdot 10^{12}$ FLOPS at IBM Blue Gene level of efficiency

What can be computed fairly easily in universal algebra with such a resource?

A database of finite structures

In 2003 I started a list of **varieties** and **quasivarieties**

to collect some basic information about them

The list is still very much **under construction**

Current version is limited by the storage format (wiki pages)

Difficult to use and extend the information within a computer algebra system

New version: use a **declarative** data format

that is **human-readable** and **machine-readable**

Should integrate well with **web browsers** (via JavaScript)

automated theorem provers such as Prover9/Mace4

and computer packages such as **Sage** and **UACalc** (via Python)

Each **(quasi)variety** is considered as a **category**

Around 100000 smallest members up to isomorphism are computed

Also compute **generators** for the **morphisms** between objects

Requires computing all **maximal proper subalgebras**

all **maximal proper homomorphic images** of each algebra

and their isomorphisms to other objects

Simple example

The **category of sets**: Objects (up to **isomorphism**) are

$\mathbf{0} = \emptyset, \mathbf{1} = \{0\}, \mathbf{2} = \{0, 1\}, \ldots, \mathbf{n} = \{0, 1, \ldots, n-1\}, \ldots$

A function $f : \mathbf{n} \to \mathbf{m}$ is given by $[f(0), f(1), \ldots, f(n-1)]$

Generators for the morphisms: $[\,] : \emptyset \to \{0\}$

$[1] : \{0\} \to \{0, 1\}$ and $[0, 0] : \{0, 1\} \to \{0\}$

$[1, 2] : \{0, 1\} \to \{0, 1, 2\}$ and $[0, 1, 0] : \{0, 1, 2\} \to \{0, 1\}$

$f_n : \mathbf{n} \to \mathbf{n}{+}1$ where $f_n(i) = i + 1$

$g_n : \mathbf{n}{+}1 \to \mathbf{n}$ where $g_n(i) = i$ if $i < n$ and $g_n(n) = 0$

And the transposition $(01) = [1, 0, 2, 3, \ldots, n-1] : \mathbf{n} \to \mathbf{n}$

Lemma: All other morphisms are **compositions** of these

Proof:
$Aut(\mathbf{n}) = S_n$ is generated by (01) and $(012\ldots n-1) = g_n \circ f_n$

Let $h : \mathbf{n} \to \mathbf{m}$ be any function and let $k = |f[\mathbf{n}]|$

Then $h = f \circ g$ where

$g : \mathbf{n} \to \mathbf{k}$ is **surjective** and $f : \mathbf{k} \to \mathbf{m}$ is **injective**

$g = g_k \circ p_1 \circ g_{k+1} \circ p_2 \circ \cdots \circ p_{n-k-1} \circ g_{n-1}$ and

$f = q \circ f_{m-1} \circ f_{m-2} \circ \cdots \circ f_k$ for some permutations p_i, q $\square$

Recall that the **skeleton** of a category is obtained by choosing one object of each isomorphism class and all morphisms between these objects

So we represent the **skeleton** of each category

The **subdirectly irreducible** members of an algebraic category are the objects that have exactly **one maximal proper homomorphic image**

The HS-poset of a variety is defined by $A \leq_{HS} B$ if $A \in HS(B)$

For **congruence distributive varieties** the lattice of **finitely generated subvarieties** is given by the finite order ideals of the HS-poset of subdirectly irreducibles

The category of Boolean algebras

We quickly run into a **problem** if we want to store the 100000 smallest Boolean algebras

Often it is more efficient to move to a **dual category** in which the objects and morphisms are easier to handle

For a finite Boolean algebra, the dual is the **set** of **atoms**

So we already solved this: use the **category of sets**

In general, use the theory of **natural dualities** that Brian Davey developed and presented at the NATO Institute of Advanced Studies Summer School

The category of distributive lattices

Up to isomorphism there are

$$1+1+1+2+3+5+8+15+26+47+82+151+269+494+891+1639+2978+5483+10006+18428+33749+62162 = 136441$$

distributive lattices of size up to 22

Could easily represent them directly

But it is much more **efficient** to use the Priestley duals:

136441 finite posets with **order-preserving maps**

What to use as **generators** for this category?

Again, use **generators** for the **automorphism groups**

and duals of **maximal embeddings and hom. images**

Which **orderpreserving maps** are **dual** to these?

[**Adams, Dwinger, Schmid 1996**] Maximal sublattices of finite distributive lattices

Use orderpreserving maps between posets that have the **same size** and where a minimal number of **incomparable elements** are mapped to **comparable elements**

These maps correspond to **covers** in the poset of partial orders

Also use epimorphisms from $n+1$-chains to n-chains and embeddings from any poset P to $P \cup \{*\}$ where $*$ is a new incomparable element

The format of the database

1. A list of **first-order theories** (mostly **varieties**)

2. For each theory in the list, a list of **smallest finite models** of the theory with **morphism generators** between them

The compressed size of the lists in 2. should be less than a few hundred MBytes

The entries for 1. are in the following format:

{"id": "short name", "name": "Long name",
"defn": "detailed English definition",
"signature": {"LaTeXsymbol": [arity,"infixl"(,priority)], ...}
"bgtheory": "background theory selected",
"axioms": ["axiom1 (in LaTeX)", "axiom2", ...],
"nmodels": [1, ..., number of models of size n, ...],
"properties": {"property name": value, ...},
"subclasses": ["shortname for max subclass", ...] },

{“id”: “DLat”, “name”: “Distributive lattices”,
“defn”: “lattices with meet distributing over join (or equivalently join distributing over meet)”,
“signature”: {“\vee”:[2,“infixl”,60], “\wedge”:[2,“infixl”,60]},
“axioms”: [“(x\vee y)\vee z = x\vee (y\vee z)”, “x\vee y = y\vee x”, “x\vee x = x”, “(x\wedge y)\wedge z = x\wedge(y\wedge z)”, “x\wedge y = y\wedge x”, “x\wedge x = x”, “x\wedge(x\vee y) = x = x\vee(x\wedge y)”, “x\wedge(y\vee z) = (x\wedge y)\vee(x\wedge z)”],
“nmodels”: [1, 1, 1, 2, 3, 5, 8, 15, 26, 47, 82, 151, 269, 494, 891, 1639, 2978, 5483, 10006, 18428, 33749, 62162, ..., 908414736485],
“properties”: {“Classtype”: “variety”, “QEqTheory”: “decidable”, “FOTheory”: “undecidable”, “CD”: “yes”, “CP”: “no”, “CR”: “no”, “CU”: “no”, “CEP”: “yes”, “EDPC”: “yes”, “AP”: “yes”, “SAP”: “no”, “ES”: “no”, “LF”: “yes”, “RS”: “2”},
“superclasses”: [“MLat”, “SDLat”],
“subclasses”: [“BDLat”, “BrouwA”, “DRL”] },

Categories of mathematical structures

Home | Alphabetical | Information | Contact | ☑ Axioms | ☐ Model counts

Algebraic Structures

1. Sets (**Set**): sets with no operations ()
2. Monounary algebras (**Alg(1)**): sets with a unary operation ()
3. Duounary algebras (**Alg(1,1)**): sets with two unary operations ()
4. Binars (**Alg(2)**): sets with a binary operation (above **Sgrp, CBin, IBin**)
    5. Commutative binars (**CBin**): sets with a commutative binary operation (below **Alg(2)**, above **CSgrp, CIBin, λLat**)
       Axioms: $xy = yx$
    6. Idempotent binars (**IBin**): sets with an idempotent binary operation (below **Alg(2)**, above **Bnd, CIBin, Drctd**)
       Axioms: $xx = x$
        7. Commutative idempotent binars (**CIBin**): sets with an idempotent binary operation (below **CBin, IBin**, above **CDrctd, Sla**
           Axioms: $xy = yx, xx = x$
    8. Semigroups (**Sgrp**): sets with an associative binary operation (below **Alg(2)**, above **Bnd, CSgrp, Mon**)
       Axioms: $(xy)z = x(yz)$
        9. Commutative semigroups (**CSgrp**): semigroups with a commutative operation (below **CBin, Sgrp**, above **Slat**)
           Axioms: $(xy)z = x(yz), xy = yx$
        10. Bands (**Bnd**): semigroups with an idempotent operation (below **IBin, Sgrp**, above **Slat, SkLat**)
            Axioms: $(xy)z = x(yz), xx = x$
            11. Semilattices (**Slat**): semigroups with an idempotent operation (below **Bnd, CIBin, CSgrp, CDrctd**, above **USlat**)
                Axioms: **Bnd** and $xy = yx$
        12. Monoids (**Mon**): semigroups expanded with an identity element (below **Sgrp**, above **CMon, Grp, IMon, RL**)
            Axioms: $(xy)z = x(yz), x1 = x = 1x$
            13. Commutative moniods (**CMon**): monoids with a commutative binary operation (below **Mon**, above **AbGrp, MV, USlat**

59. Heyting algebras (**HA**): relatively pseudocomplemented bounded distributive lattices (below **BDLat, BrouwA**, above **GAlg**)
    Axioms: **BDLat** and $x \to x = 1, x \wedge (x \to y) = x \wedge y, (x \to y) \wedge y = y,$
    $x \to (y \wedge z) = (x \to y) \wedge (x \to z)$
    60. Gödel algebras (**GAlg**): prelinear Heyting algebras, i.e. subdirectly irreducibles are linear (below **HA**, above **BA**)
        Axioms: **HA** and $(x \to y) \vee (y \to x) = 1$
        61. Boolean algebras (**BA**): complemented distributive lattices (below **GAlg, MV**, above **ModalA**)
            Axioms: **DLat** and $x \vee \neg x = 1, x \wedge \neg x = 0$
            62. Modal algebras (**ModalA**): Boolean algebras with a unary operation that distributes over all finite joins (below **BA**, above **CloA**)
                Axioms: **BA** and $f(x \vee y) = f(x) \vee f(y), f(0) = 0$
                63. Closure algebras (**CloA**): Modal algebras where the operator is increasing and idempotent (below **ModalA**, above **MondcA**)
                    Axioms: **ModalA** and $x \leq f(x), f(f(x)) = f(x)$
                    64. Monadic algebras (**MondcA**): Closure algebras where the operator commutes with complementation (below **CloA**, above **Triv**)
                        Axioms: **CloA** and $\neg f(x) = f(\neg x)$
                        65. Trivial algebras (**Triv**): algebras with exactly one element (below **MondcA, BoolGrp, Dio, Fld, USlat**,)
                            Axioms: $x = y$

Format for algebras and relational structures

“id”: { “cardinality”: 2,

“operations”: {“\cdot”:[[0,0],[0,1]], “1”:1, ...},

“relations”: {“\le”:[[1,1],[0,1]], “\prec”:{0:[1],1:[]}, ...},

“names”: {0: “\bot”, 1: “\top”},

“positions”: [[x1,y1], [x2,y2], ...],

“properties”: {“P”: “True”, “Q”: “False”, ...},

“autgens”: [g1, g2, ...],

“maxsubs”: [[id1,[...]], [id2,[...]], ...],

“maximages”: [[id3,[...]], [id4,[...]], ...] },

Semirings

A **semiring** is an algebra $(S, +, \cdot)$ such that

$$(x + y) + z = x + (y + z), \quad x + y = y + x$$

$$(xy)z = x(yz), \quad x(y+z) = xy+xz \quad \text{and} \quad (x+y)z = xz+yz$$

It is **simple** if it has only two congruences

Theorem: [Monico 2004] A finite simple semiring S is either

- a ring or
- is **idempotent** ($x + x = x$ for all $x \in S$) or
- $(S, \cdot)$ is a simple semigroup with absorbing element ∞ and $S + S = \infty$

Idempotent semirings are join-semilattices with $\cdot$ joinpreserving

Idem. semirings of size n: [1, 6, 61, 866, **15751, 354409**]

Simple idem. semirings of size n: [1, 6, **3, 1, 4, 3**]

Example: For a join-semilattice L the set End(L) is an idempotent semiring under pointwise join and composition, with id_L as identity

A semiring has a **neutral element** 0 if $x + 0 = x$

It has a **zero** if this element also satisfies $0x = 0 = x0$

Idem. semirings with neutral 0: [1, 6, 44, 479, 6738, ...]

Idem. semirings with a zero: [1, 2, 10, 68, 520, 4447 ...]

Idem. semirings with 1 and zero: [1, 1, 3, 20, 149, 1488, **18554, 295292**]

If L has a **bottom** element, then End(L) always has a **zero**

Zumbrägel [2008] classified **all** finite simple idempotent semiring with zero as **dense** subsemirings of End(L) where L is a join-semilattice with bottom

[Dense means it contains all maps $e_{a,b}(x) = b$ if $x \nleq a$ and 0 otherwise]

Kendziorra [2012] extended this classification to simple semirings with a neutral element

Full classification of finite simple semirings is still open

Computation of simple idempotent semirings **without neutral elements** is an ongoing project

Constructing all modular lattices of size n

Joint work with **Nathan Lawless** (Chapman University)

Heitzig, Reinhold [2002] enumerated all lattices up to size 18

Erne, Heitzig, Reinhold [2002] enumerated all distributive lattices up to size 49

By 2008 modular lattices had only been counted up to size 11:

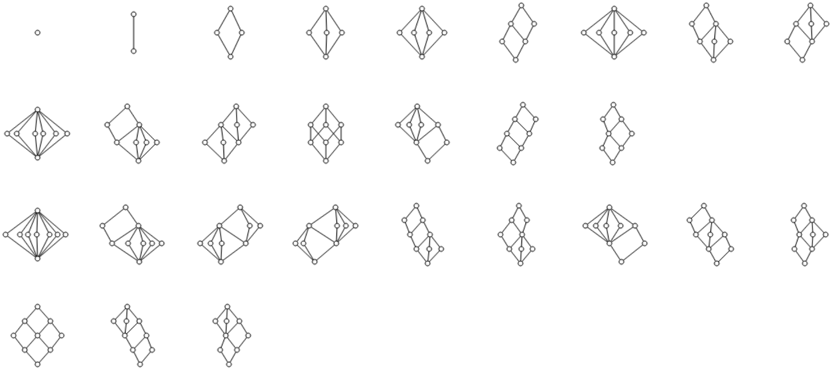
n	1	2	3	4	5	6	7	8	9	10	11
m_n	1	1	1	2	4	8	16	34	72	157	343

where m_n is the number of modular lattices of size n

Belohlavek and Vychodil [2009] showed that $m_{12} = 766$

Modular lattices up to size 9

The first few vertically indecomposable modular lattices



Using a cluster of 64 processors at a costs of about $10 a day

[J. and Lawless 2013]:

n	13	14	15	16	17	18
m_n	1718	3899	8898	20475	47321	110024

n	19	20	21	22	23	24
m_n	256791	601991	1415768	3340847	?	?

The calculations use B. McKay's **nauty** program to find automorphism generators and eliminate isomorphic copies

Faigle and Herrmann [1981] axiomatized poset geometries that are dual to modular lattices

These duals may be easier to enumerate

n	All lattices	Semimodular	Modular	V. I. Mod	Distrib	S. I. Lat	SI Mod
6	15	8	8	2	5	4	1
7	53	17	16	3	8	16	1
8	222	38	34	7	15	69	2
9	1,078	88	72	12	26	360	3
10	5,994	**212**	157	28	47	**2,103**	4
11	37,622	**530**	343	54	82	**13,867**	7
12	262,776	**1,376**	766	127	151	**100,853**	15
13	2,018,305	**3,693**	**1,718**	**266**	269		**28**
14	16,873,364	**10,232**	**3,899**	**614**	494		**53**
15	152,233,518	**29,231**	**8,898**	**1,356**	891		**106**
16	1,471,613,387	**85,906**	**20,475**	**3,134**	1,639		**226**
17	15,150,569,446	**259,291**	**47,321**	**7,091**	2,978		**479**
18	165,269,824,761	**802,308**	**110,024**	**16,482**	5,483	←Erne	
19	↑Heitzig &	**2,540,635**	**256,791**	**37,929**	10,006	Heitzig	
20	Reinhold 2002	**8,220,218**	**601,991**	**88,622**	18,428	Reinhold	
21		**27,134,483**	**1,415,768**	**206,295**	33,749	2002 up	
22	Bold entries '13	J. & Lawless	**3,340,847**	**484,445**	62,162	to n=49	

Enumerating lattice contexts

Formal Concept Analysis connects binary relations (contexts) with complete lattice using Birkhoff's polarities

Every finite lattice L has a unique **reduced context** given by $\leq$ restricted to $J(L) \times M(L)$

Recover L as the lattice of **Galois closed sets** of the context

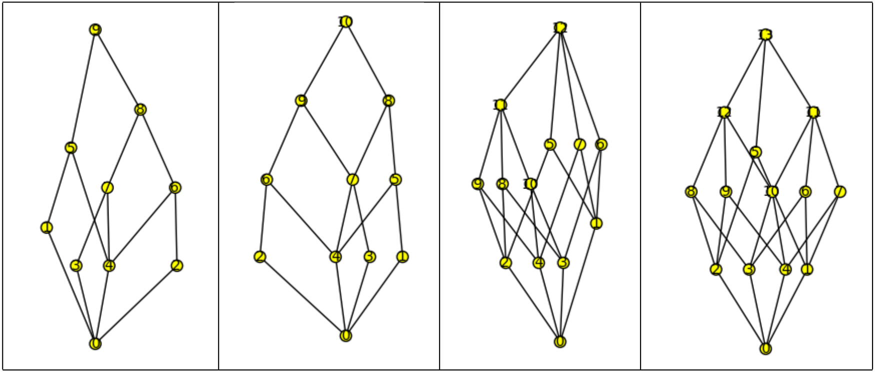
How many reduced contexts are there from m to n elements?

Number of reduced contexts with $m + n$ elements

- means there is no context with this combination of m, n

$m \backslash n$	1	2	3	4	5	6	7	8
1	1	-	-	-	-	-	-	-
2	-	2	-	-	-	-	-	-
3	-	-	7	2	-	-	-	-
4	-	-	2	45	50	25	4	-
5	-	-	-	50	717	2241	3670	3598
6	-	-	-	25	2241	37535	266178	
7	-	-	-	4	3670	266178		
8	-	-	-	-	3598			

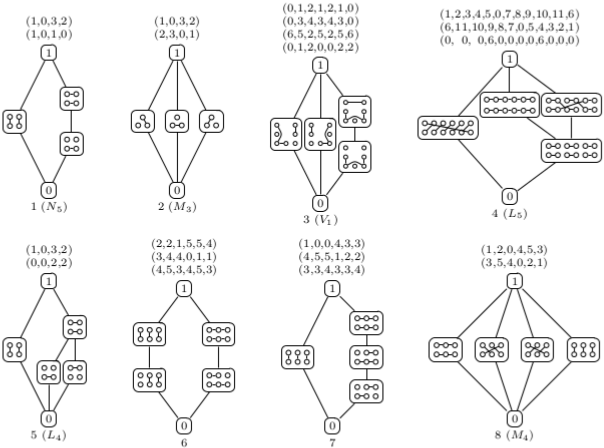
The calculation used Brendan McKay's bipartite graph generator **genbg**



Finite lattice representation problem

Constructing finite algebras with prescribed small congruence lattices

Joint work with W. DeMeo, R. Freese, B. Lampe, J.B. Nation



Made a list of 7-element lattices, removed the distributive ones

We removed vertically and horizontally decomposable ones

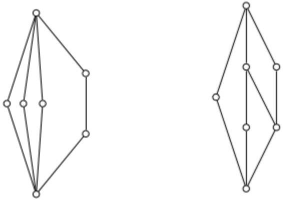
Wrote programs to search for closed representations in Equ(n)

Used GAP to search for intervals in subgroup lattices

We got down to 2 interesting cases

The first one lead to the development of overalgebras

The second one is still open



Latest version of the database

math.chapman.edu/~jipsen/mathstructures

also in a Git repository on GitHub

(obviously still under construction...)

Conclusion (moral of the story)

If your algorithm has exponential complexity

that doesn't mean its useless

Just wait a couple of years and you can do the next step

for **the same cost** as the previous step

Ten dollars a day can go a long way!

Some References

- D. M. Clark and B. A. Davey, Natural dualities for the working algebraist, Cambridge Studies in Advanced Mathematics 57, 1998.
- B. A. Davey, Duality theory on ten dollars a day, in *Algebras and Orders*, (I. G. Rosenberg and G. Sabidussi, eds), Kluwer Academic Publishers, 1993, 71–111.
- R. Freese, E. Kiss and M. Valeriote, Universal Algebra Calculator, www.uacalc.org
- P. Jipsen, Mathematical Structures, `math.chapman.edu/~jipsen/structures`
- W. McCune, Prover9 and Mace4, www.cs.unm.edu/~mccune/Prover9, 2005-2010.
- W. A. Stein et al., Sage Mathematics Software (Version 5.6), The Sage Development Team, 2012, www.sagemath.org

Thank You

BLAST 2013, August 5-9, Chapman University, Orange, CA