The Universal Algebra Calculator UACalc

Ralph Freese

The Universal Algebra Calculator

- Short name: UACalc

The Universal Algebra Calculator

- Short name: UACalc
- Download the program from the website.
- The website: `http://uacalc.org/`

The Universal Algebra Calculator

- Short name: UACalc
- Download the program from the website.
- The website: `http://uacalc.org/`
- Contributors:
  - Matt Valeriote and some of his students
  - Emil Kiss
  - Mike Behrish
  - William DeMeo

Tutorial: Outline

- Getting algebras into the program. From the File menu:
  - Built-in
  - New
  - Open

Tutorial: Outline

- Getting algebras into the program. From the File menu:
  - Built-in
  - New
  - Open
- Adding, Changing, Deleting operations.

Tutorial: Outline

- Getting algebras into the program. From the File menu:
  - Built-in
  - New
  - Open
- Adding, Changing, Deleting operations.
- The Current Algebra

Tutorial: Outline

- Getting algebras into the program. From the File menu:
  - Built-in
  - New
  - Open
- Adding, Changing, Deleting operations.
- The Current Algebra
- Basic constructions: Con, Sub, Drawing, HSP.
- New Element Key Table
- Using of the drawing tools

Exercises

- Choose File $\rightarrow$ New. Make a 5 element algebra.
- Add a binary operation. Choose "Random" for the default element.
- Choose Tasks $\rightarrow$ Primality to see if this algebra is primal.
- Click the Editor tab. Try to change something so the answer is the opposite.
- Hint: When editing a cell in the table, you need to move out of the cell to register the change.
- Choose File $\rightarrow$ Built in Algs and load polin, lyndon, m3, n5.
- Switch algebras (at the bottom) to lyndon.

Exercises Continued

- Switch algebras (at the bottom) to lyndon.
- Go to the Con tab and click Go.
- Play with the controls to see what the do. Click on some elements.
- Note there are two atoms that are meet irreducible.
- Click on one of them and then right click on it and make the quotient algebra.
- Make that the current algebra (at the bottom) and switch to the Editor tab. Check it is not editable but there is a translation table for the elements.
- Choose Tasks → B in V(A) ? and show lyndon is in the variety of the quotient.
- So this 6 element algebra is also not finitely based. (Howard Lee).

Computations Tab and Background Tasks

- Regions: Description, Results Table, Task List, Details

Computations Tab and Background Tasks

- Regions: Description, Results Table, Task List, Details
- Taks List: switching, canceling, deleting.

Computations Tab and Background Tasks

- Regions: Description, Results Table, Task List, Details
- Taks List: switching, canceling, deleting.
- Free Algebras
  - use all coordinates
  - thin coordinates
  - thin and decompose

Computations Tab and Background Tasks

- Regions: Description, Results Table, Task List, Details
- Taks List: switching, canceling, deleting.
- Free Algebras
  - use all coordinates
  - thin coordinates
  - thin and decompose
- Results Table: saving as a CSV file.

Computations Tab and Background Tasks

- Regions: Description, Results Table, Task List, Details
- Taks List: switching, canceling, deleting.
- Free Algebras
  - use all coordinates
  - thin coordinates
  - thin and decompose
- Results Table: saving as a CSV file.
- Test if $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

More Exercises

- Switch to m3.
- Make the 3-generated free algebra using each of the options.
- Try to figure the difference is.
- Switch to one of these free algebras and to the Editor tab.
- This is essentially the Birkhoff basis.
- Compute $\mathbf{F}_{\mathbf{M}_3}(4)$. Takes about 10 minutes. It runs in the background.
- Find $\mathbf{F}_{\mathbf{N}_5}(3)$.
- Switch algebras to $\mathbf{F}_{\mathbf{N}_5}(3)$ and draw it.

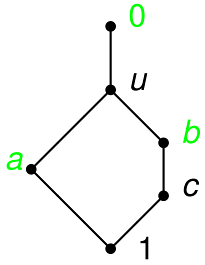
Directoids: Ježek and Quackenbush

A **directoid** is a groupoid defined on a p. o. set such that

$$x \leq xy \qquad y \leq xy \qquad x \leq y \implies xy = yx = y$$

It is an equational class:

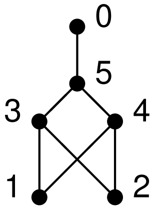
- Is every finite directoid finitely based?
- Hajilarov gave a 6 element directoid, **H**, which he asserted is INFB:



$\cdot$	1	a	b	c	u	0
1	1	a	b	c	u	0
a	a	a	0	u	u	0
b	b	0	b	b	u	0
c	c	u	b	c	u	0
u	u	u	u	u	u	0
0	0	0	0	0	0	0

Directoids: Kate Scott Owens

The directoid **D**:

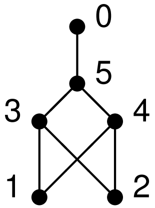


·	1	2	3	4	5	0
1	1	3	3	4	5	0
2	3	2	3	4	5	0
3	3	3	3	0	5	0
4	4	4	0	4	5	0
5	5	5	5	5	5	0
0	0	0	0	0	0	0

The argument that **H** is INFB implies $\mathbf{D} \in \boldsymbol{V}(\mathbf{H})$.

Directoids: Kate Scott Owens

The directoid **D**:

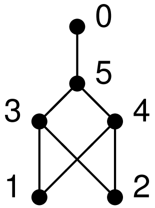


·	1	2	3	4	5	0
1	1	3	3	4	5	0
2	3	2	3	4	5	0
3	3	3	3	0	5	0
4	4	4	0	4	5	0
5	5	5	5	5	5	0
0	0	0	0	0	0	0

The argument that **H** is INFB implies $\mathbf{D} \in \boldsymbol{V}(\mathbf{H})$.
But it's not.

Directoids: Kate Scott Owens

The directoid **D**:



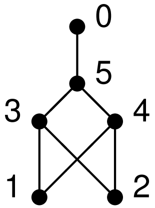
$\cdot$	1	2	3	4	5	0
1	1	3	3	4	5	0
2	3	2	3	4	5	0
3	3	3	3	0	5	0
4	4	4	0	4	5	0
5	5	5	5	5	5	0
0	0	0	0	0	0	0

The argument that **H** is INFB implies $\mathbf{D} \in \boldsymbol{V}(\mathbf{H})$.
But it's not. The calculator gives the equation

$$x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$$

Directoids: Kate Scott Owens

The directoid **D**:



·	1	2	3	4	5	0
1	1	3	3	4	5	0
2	3	2	3	4	5	0
3	3	3	3	0	5	0
4	4	4	0	4	5	0
5	5	5	5	5	5	0
0	0	0	0	0	0	0

The argument that **H** is INFB implies $\mathbf{D} \in \boldsymbol{V}(\mathbf{H})$.
But it's not. The calculator gives the equation

$$x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$$

and claims it holds in **H** and fails in **D** under the substitution

$$x_0 \mapsto 1 \quad x_1 \mapsto 2 \quad x_2 \mapsto 4 \quad x_3 \mapsto 5$$

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of **B**.

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of **B**.
- Start calculating $\mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) = \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(x_0, \ldots, x_{k-1})$, keeping

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of $\mathbf{B}$.
- Start calculating $\mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) = \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(x_0, \ldots, x_{k-1})$, keeping
  - A map from the elements to the term that gave them.

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of $\mathbf{B}$.
- Start calculating $\mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) = \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(x_0, \ldots, x_{k-1})$, keeping
  - A map from the elements to the term that gave them.
  - A partial homomorphism from $\varphi : \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) \to \mathbf{B}$.

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of **B**.
- Start calculating $\mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) = \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(x_0, \ldots, x_{k-1})$, keeping
  - A map from the elements to the term that gave them.
  - A partial homomorphism from $\varphi : \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) \to \mathbf{B}$.
- If

  $$f(a_0, \ldots, a_{r-1}) = a$$

  is new, then

  $$t_a = f(t_{a_0}, \ldots, t_{a_{r-1}}) \quad \text{and}$$
  $$\varphi(a) = f(\varphi(a_0), \ldots, \varphi(a_{r-1}))$$

Testing $\mathbf{B} \in \boldsymbol{V}(\mathbf{A})$

- Find a minimal sized generating set $\{g_0, \ldots, g_{k-1}\}$ of **B**.
- Start calculating $\mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) = \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(x_0, \ldots, x_{k-1})$, keeping
  - A map from the elements to the term that gave them.
  - A partial homomorphism from $\varphi : \mathbf{F}_{\boldsymbol{V}(\mathbf{A})}(k) \to \mathbf{B}$.
- If $a = f(a_0, \ldots, a_{r-1})$ is **not** new, and

$$\varphi(a) \neq f(\varphi(a_0), \ldots, \varphi(a_{r-1}))$$

then the equation (of the Birkhoff basis):

$$t_a \approx f(t_{a_0}, \ldots, t_{a_{r-1}})$$

fails in **B** under the substitution $x_i \mapsto g_i$.

$\mathbf{D} \notin \boldsymbol{V}(\mathbf{H})$

- $x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$ witnesses this (under 1 second).

$\mathbf{D} \notin \boldsymbol{V}(\mathbf{H})$

- $x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$ witnesses this (under 1 second).
- $|\mathbf{F}_{\boldsymbol{V}(\mathbf{H})}(4)| = 26,467$ (60 minutes)

$\mathbf{D} \notin \boldsymbol{V}(\mathbf{H})$

- $x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$ witnesses this (under 1 second).
- $|\mathbf{F}_{\boldsymbol{V}(\mathbf{H})}(4)| = 26,467$ (60 minutes)
- So the Birkhoff basis has over 700 million equations.

$\mathbf{D} \notin \boldsymbol{V}(\mathbf{H})$

- $x_3((x_0x_1)(x_0(x_1x_2))) \approx (x_0x_1)(x_3(x_0(x_1x_2)))$
  witnesses this (under 1 second).
- $|\mathbf{F}_{\boldsymbol{V}(\mathbf{H})}(4)| = 26,467$ (60 minutes)
- So the Birkhoff basis has over 700 million equations.
- Testing $\mathbf{H} \in \boldsymbol{V}(\mathbf{H})$ takes about 80 minutes.

Programming and Javadoc

- The page `http://uacalc.org/download/` gives instructions for getting the source code.
- The page `http://uacalc.org/doc/` documents the methods.
- Try the CongruenceLattice link on the lower left.
- Note the method `commutator(BinaryRelation S, BinaryRelation T)`.
- You can use this in your code or with the command line interface:

Command Line

```
ralph@mahiloa:~/UACalc/UACalc_CLI]$ uacalc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Welcome to the command line version of UACalc!
    to exit type quit()
    (more help coming)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
>>> f3 = AlgebraIO.readAlgebraFile("/home/ralph/UACalc/Algebras/f3.ua")
>>> f3.cardinality()
3
>>> conlat = f3.con().getUniverseList()
>>> conlat
[|0|1|2|, |0|1,2|, |0,1,2|]
>>> theta = conlat[1]
>>> theta
|0|1,2|
>>> one = conlat[2]
>>> one
|0,1,2|
>>> f3.con().commutator(theta,one)
|0|1|2|
>>> f3.con().commutator(one,theta)
|0|1,2|
>>> quit()
```