UCALC and testing for Maltsev conditions

Matt Valeriote

McMaster University

22 May, 2018

Dedication

This talk is dedicated to Bill, J.B., and Ralph for the example that they set by embracing the computer, way back in the mid 1980's, as a vital tool for conducting research in our field.

History and acknowledgments

The first version of the program has been written by students of Matt Valeriote in 1988-91, for XWindows. This version had many of the features of the present program, and some additional ones, too, which we plan to implement. These include working with terms and identities, drawing subalgebra lattices, etc. However that version has many bugs. It is available from

ftp://icarus.math.mcmaster.ca:/pub/UA/Algebra.tar.gz

Many of the ideas for writing the present program came from that version. This old version had some partial MS-DOS ports, but these are now made obsolete by the present program.

The other source of inspiration has been Ralph Freese's lattice drawing program, which is integrated into the present program. It can display any lattice, not just congruence lattices. You can try out the original version online, or download the source. The address is

http://www.math.hawaii.edu/~ralph/LatDraw

The program makes use of Ralph Freese's algorithms concerning partitions, to speed up calculations. You can download the corresponding reprints from

http://www.math.hawaii.edu/~ralph/papers.html

** From `http://www.math.hawaii.edu/~ralph/software/uaprog/oldversion.html`

Testing for a near unanimity term

- The algebra demo1.ua is a 4 element algebra that has a 5-ary near unanimity operation as a basic operation.

Testing for a near unanimity term

- The algebra demo1.ua is a 4 element algebra that has a 5-ary near unanimity operation as a basic operation.
- It might also, incidentally, have a 4-ary near unanimity term operation.

Testing for a near unanimity term

- The algebra demo1.ua is a 4 element algebra that has a 5-ary near unanimity operation as a basic operation.
- It might also, incidentally, have a 4-ary near unanimity term operation.
- Question: How can we determine this?

Testing for a near unanimity term

- The algebra demo1.ua is a 4 element algebra that has a 5-ary near unanimity operation as a basic operation.
- It might also, incidentally, have a 4-ary near unanimity term operation.
- Question: How can we determine this?
- Answer: Generate $\mathbb{F}(x,y)$, the 2-generated free algebra in $\mathbf{V}(demo1)$ and then look for the tuple (x,x,x,x) in the subalgebra of $\mathbb{F}(x,y)^4$ generated by

$$\{(y,x,x,x),(x,y,x,x),(x,x,y,x),(x,x,x,y)\}$$

Testing for a near unanimity term

- The algebra demo1.ua is a 4 element algebra that has a 5-ary near unanimity operation as a basic operation.
- It might also, incidentally, have a 4-ary near unanimity term operation.
- Question: How can we determine this?
- Answer: Generate $\mathbb{F}(x, y)$, the 2-generated free algebra in $\mathbf{V}(demo1)$ and then look for the tuple (x, x, x, x) in the subalgebra of $\mathbb{F}(x, y)^4$ generated by

$$\{(y, x, x, x), (x, y, x, x), (x, x, y, x), (x, x, x, y)\}$$

- UACALC can do this for us.

Is there a better way?

Motivation

Is there a better way?

Motivation

- When testing some hypotheses or conjectures, one sometimes will need to build algebras that satisfy certain Maltsev conditions, but not others.

Is there a better way?

Motivation

- When testing some hypotheses or conjectures, one sometimes will need to build algebras that satisfy certain Maltsev conditions, but not others.
- The previous algebra was constructed sometime in 2004 while working on special cases of the Constraint Satisfaction Problem.

Is there a better way?

Motivation

- When testing some hypotheses or conjectures, one sometimes will need to build algebras that satisfy certain Maltsev conditions, but not others.
- The previous algebra was constructed sometime in 2004 while working on special cases of the Constraint Satisfaction Problem.
- Usually, the algebras were designed to satisfy a given Maltsev condition, but it wasn't so clear if they also, incidentally, satisfied others.

Is there a better way?

Motivation

- When testing some hypotheses or conjectures, one sometimes will need to build algebras that satisfy certain Maltsev conditions, but not others.
- The previous algebra was constructed sometime in 2004 while working on special cases of the Constraint Satisfaction Problem.
- Usually, the algebras were designed to satisfy a given Maltsev condition, but it wasn't so clear if they also, incidentally, satisfied others.
- Back then, the only algorithms that were implemented by UACALC involved building free algebras, and so could not be used for most algebras, especially those that had rich local structure.

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO$'$) to the given problem.

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO$'$) to the given problem.
- An instance of GEN-CLO$'$consists of

No!

- In the paper “On the complexity of some Maltsev conditions”, Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO$'$) to the given problem.
- An instance of GEN-CLO$'$consists of
  - a finite set A,

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO$'$) to the given problem.
- An instance of GEN-CLO$'$consists of
  - a finite set A,
  - a finite set of finitary operations $\mathcal{F}$ on A, and

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO$'$) to the given problem.
- An instance of GEN-CLO$'$consists of
  - a finite set A,
  - a finite set of finitary operations $\mathcal{F}$ on A, and
  - a unary function $h : A \to A$.

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO′) to the given problem.
- An instance of GEN-CLO′consists of
  - a finite set A,
  - a finite set of finitary operations $\mathcal{F}$ on A, and
  - a unary function $h : A \to A$.
- The problem is to decide if h is in the clone generated by $\mathcal{F}$.

No!

- In the paper "On the complexity of some Maltsev conditions", Ralph Freese and I were able to prove that testing for many familiar Maltsev conditions is hard in general.
- The hardness is obtained by reducing the EXP-TIME complete Clone Membership Problem (GEN-CLO′) to the given problem.
- An instance of GEN-CLO′consists of
  - a finite set A,
  - a finite set of finitary operations $\mathcal{F}$ on A, and
  - a unary function $h : A \to A$.
- The problem is to decide if h is in the clone generated by $\mathcal{F}$.
- This decision problem was shown to be hard by Bergman, Juedes, and Slutzki in the paper "Computational complexity of term-equivalence". Harvey Friedman is also credited with proving this result.

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

- *Is* $\mathbf{V}(\mathbb{A})$ *congruence modular?*

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

- *Is* $\mathbf{V}(\mathbb{A})$ *congruence modular?*
- *Is* $\mathbf{V}(\mathbb{A})$ *congruence distributuve?*

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO′such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

- *Is $\mathbf{V}(\mathbb{A})$ congruence modular?*
- *Is $\mathbf{V}(\mathbb{A})$ congruence distributuve?*
- *Is $\mathbf{V}(\mathbb{A})$ congruence (meet) semidistributive?*

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

- *Is* $\mathbf{V}(\mathbb{A})$ *congruence modular?*
- *Is* $\mathbf{V}(\mathbb{A})$ *congruence distributuve?*
- *Is* $\mathbf{V}(\mathbb{A})$ *congruence (meet) semidistributive?*
- *Does* $\mathbf{V}(\mathbb{A})$ *omit the unary type?*

A hardness result

- We provide a way to construct a finite algebra $\mathbb{A}_I$ from an instance I of GEN-CLO$'$such that:
  - If I is a **yes** instance, then $\mathbb{A}_I$ has a number of interesting idempotent term operations, such as Jónsson terms, and
  - If I is a **no** instance, then $\mathbb{A}_I$ has no non-trivial idempotent terms.
- Using this "all-or-nothing" construction we proved the following theorem.

Theorem (Freese-Val.)

Let $\mathbb{A}$ be a finite algebra. The following problems are EXP-TIME complete:

- *Is* $\mathbf{V}(\mathbb{A})$ *congruence modular?*
- *Is* $\mathbf{V}(\mathbb{A})$ *congruence distributuve?*
- *Is* $\mathbf{V}(\mathbb{A})$ *congruence (meet) semidistributive?*
- *Does* $\mathbf{V}(\mathbb{A})$ *omit the unary type?*
- *Does* $\mathbf{V}(\mathbb{A})$ *omit the unary and semilattice types?*

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.
- He provided a polynomial-time algorithm to decide if a given finite idempotent algebra $\mathbb{A}$ generates a variety that omits the unary type.

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.
- He provided a polynomial-time algorithm to decide if a given finite idempotent algebra $\mathbb{A}$ generates a variety that omits the unary type.

Definition

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.
- He provided a polynomial-time algorithm to decide if a given finite idempotent algebra $\mathbb{A}$ generates a variety that omits the unary type.

Definition

- An operation $f(x_1, x_2, \ldots, x_n)$ on a set A is idempotent if for all $a \in A$, $f(a, a, \ldots, a) = a$.

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.
- He provided a polynomial-time algorithm to decide if a given finite idempotent algebra $\mathbb{A}$ generates a variety that omits the unary type.

Definition

- An operation $f(x_1, x_2, \ldots, x_n)$ on a set A is idempotent if for all $a \in A$, $f(a, a, \ldots, a) = a$.
- An algebra is idempotent if all of its basic operations are idempotent.

A glimmer of hope

- In the early days of his work on the Constraint Satisfaction Problem Dichotomy conjecture, Bulatov considered the complexity of a related problem.
- He provided a polynomial-time algorithm to decide if a given finite idempotent algebra $\mathbb{A}$ generates a variety that omits the unary type.

Definition

- An operation $f(x_1, x_2, \ldots, x_n)$ on a set A is idempotent if for all $a \in A$, $f(a, a, \ldots, a) = a$.
- An algebra is idempotent if all of its basic operations are idempotent.

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. Then $\mathbf{V}(\mathbb{A})$ omits the unary type if and only if some 2-generated subalgebra of $\mathbb{A}$ has a 2-element quotient that is term equivalent to a set.

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. There are polynomial-time tests to determine if:

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. There are polynomial-time tests to determine if:

- $\mathbf{V}(\mathbb{A})$ *is congruence semidistributive,*

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. There are polynomial-time tests to determine if:

- $\mathbf{V}(\mathbb{A})$ *is congruence semidistributive,*
- $\mathbf{V}(\mathbb{A})$ *is congruence meet semidistributive,*

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. There are polynomial-time tests to determine if:

- $\mathbf{V}(\mathbb{A})$ *is congruence semidistributive,*
- $\mathbf{V}(\mathbb{A})$ *is congruence meet semidistributive,*
- $\mathbf{V}(\mathbb{A})$ *satisfies a non-trivial congruence identity,*

Extensions of Bulatov's result

- By applying Bulatov's approach, and using tame congruence theoretic characterizations (i.e., local characterizations), the following theorem was proved:

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. There are polynomial-time tests to determine if:

- $\mathbf{V}(\mathbb{A})$ *is congruence semidistributive,*
- $\mathbf{V}(\mathbb{A})$ *is congruence meet semidistributive,*
- $\mathbf{V}(\mathbb{A})$ *satisfies a non-trivial congruence identity,*
- $\mathbf{V}(\mathbb{A})$ *is congruence n-permutable for some n.*

Example: Congruence meet semidistributivity

- For a finite algebra $\mathbb{A}$, $\mathbf{V}(\mathbb{A})$ is congruence meet semidistributive if this variety omits the unary and affine types.

Example: Congruence meet semidistributivity

- For a finite algebra $\mathbb{A}$, $\mathbf{V}(\mathbb{A})$ is congruence meet semidistributive if this variety omits the unary and affine types.
- When $\mathbb{A}$ is idempotent, then these types will appear in some 2-generated subalgebra of $\mathbb{A}$.

Example: Congruence meet semidistributivity

- For a finite algebra $\mathbb{A}$, $\mathbf{V}(\mathbb{A})$ is congruence meet semidistributive if this variety omits the unary and affine types.
- When $\mathbb{A}$ is idempotent, then these types will appear in some 2-generated subalgebra of $\mathbb{A}$.
- An alternate characterization that is used by UACALC to perform this test is: for all x and y in $\mathbb{A}$, if $a = (x, x)$, $b = (x, y)$ and $c = (y, x)$, then in the subalgebra $\mathbb{B}$ of $\mathbb{A}^2$ generated a, b and c

$$(a, c) \in \beta \vee (\alpha \wedge \gamma),$$

where $\alpha = \mathrm{Cg}^{\mathbb{B}}(a, c)$, $\beta = \mathrm{Cg}^{\mathbb{B}}(a, b)$, and $\gamma = \mathrm{Cg}^{\mathbb{B}}(b, c)$.

Example: Congruence meet semidistributivity

- For a finite algebra $\mathbb{A}$, $\mathbf{V}(\mathbb{A})$ is congruence meet semidistributive if this variety omits the unary and affine types.
- When $\mathbb{A}$ is idempotent, then these types will appear in some 2-generated subalgebra of $\mathbb{A}$.
- An alternate characterization that is used by UACALC to perform this test is: for all x and y in $\mathbb{A}$, if $a = (x, x)$, $b = (x, y)$ and $c = (y, x)$, then in the subalgebra $\mathbb{B}$ of $\mathbb{A}^2$ generated a, b and c

$$(a, c) \in \beta \vee (\alpha \wedge \gamma),$$

  where $\alpha = \mathrm{Cg}^{\mathbb{B}}(a, c)$, $\beta = \mathrm{Cg}^{\mathbb{B}}(a, b)$, and $\gamma = \mathrm{Cg}^{\mathbb{B}}(b, c)$.
- The runtime of the associated algorithm can be bounded be a polynomial of degree 4 in the size of $\mathbb{A}$.

Congruence modularity and distributivity

- By using the local characterizations of congruence modularity and distributivity provided by tame congruence theory, we were able to come up with polynomial-time algorithms to test for these conditions.

Congruence modularity and distributivity

- By using the local characterizations of congruence modularity and distributivity provided by tame congruence theory, we were able to come up with polynomial-time algorithms to test for these conditions.

Theorem (Hobby, McKenzie)

Let $\mathbb{A}$ be a finite algebra. Then $\mathbf{V}(\mathbb{A})$ is congruence modular (distributive) if and only if

Congruence modularity and distributivity

- By using the local characterizations of congruence modularity and distributivity provided by tame congruence theory, we were able to come up with polynomial-time algorithms to test for these conditions.

Theorem (Hobby, McKenzie)

Let $\mathbb{A}$ be a finite algebra. Then $\mathbf{V}(\mathbb{A})$ is congruence modular (distributive) if and only if

- $\mathbf{V}(\mathbb{A})$ *omits the unary and semilattice (and vector space) types, and*

Congruence modularity and distributivity

- By using the local characterizations of congruence modularity and distributivity provided by tame congruence theory, we were able to come up with polynomial-time algorithms to test for these conditions.

Theorem (Hobby, McKenzie)

Let $\mathbb{A}$ be a finite algebra. Then $\mathbf{V}(\mathbb{A})$ is congruence modular (distributive) if and only if

- $\mathbf{V}(\mathbb{A})$ *omits the unary and semilattice (and vector space) types, and*
- *all minimal sets have empty tails.*

Congruence modularity and distributivity

- By using the local characterizations of congruence modularity and distributivity provided by tame congruence theory, we were able to come up with polynomial-time algorithms to test for these conditions.

Theorem (Hobby, McKenzie)

Let $\mathbb{A}$ be a finite algebra. Then $\mathbf{V}(\mathbb{A})$ is congruence modular (distributive) if and only if

- *$\mathbf{V}(\mathbb{A})$ omits the unary and semilattice (and vector space) types, and*
- *all minimal sets have empty tails.*

Remark

We showed that in this situation, if some minimal set of some algebra in $\mathbf{V}(\mathbb{A})$ has a non-empty tail, then we could find such an algebra as a 3-generated subalgebra of $\mathbb{A}^2$.

A different approach

Remarks

A different approach

Remarks

- *In addition, we were able to prove that testing for congruence permutability and also having a majority term can be carried out using polynomial-time algorithms, in the idempotent case.*

A different approach

Remarks

- *In addition, we were able to prove that testing for congruence permutability and also having a majority term can be carried out using polynomial-time algorithms, in the idempotent case.*
- *Independently, Horowitz and McKenzie observed that our "structural" proofs could be reworked/rephrased in a more term-based manner.*

A different approach

Remarks

- *In addition, we were able to prove that testing for congruence permutability and also having a majority term can be carried out using polynomial-time algorithms, in the idempotent case.*
- *Independently, Horowitz and McKenzie observed that our "structural" proofs could be reworked/rephrased in a more term-based manner.*

Theorem

Let $\mathbb{A}$ be a finite idempotent algebra. Then $\mathbb{A}$ has a Maltsev term if and only if it has enough local Maltsev terms:
for each a, b, c, $d \in A$, there is a term operation $t_{(a,b,c,d)}(x, y, z)$ such that

$$t_{(a,b,c,d)}(a, b, b) = a \quad \text{and} \quad t_{(a,b,c,d)}(c, c, d) = d.$$

A local term algorithm

- This kind of local term result leads to a relatively easy to code polynomial-time algorithm for testing for the associated condition.

A local term algorithm

- This kind of local term result leads to a relatively easy to code polynomial-time algorithm for testing for the associated condition.
- A finite algebra $\mathbb{A}$ will have a term operation $t_{(a,b,c,d)}(x, y, z)$ such that $t_{(a,b,c,d)}(a, b, b) = a$ and $t_{(a,b,c,d)}(c, c, d) = d$ if and only if

$$(a, d) \in \mathrm{Sg}^{\mathbb{A}^2}(\{(a, c), (b, c), (b, d)\}.$$

A local term algorithm

- This kind of local term result leads to a relatively easy to code polynomial-time algorithm for testing for the associated condition.
- A finite algebra $\mathbb{A}$ will have a term operation $t_{(a,b,c,d)}(x, y, z)$ such that $t_{(a,b,c,d)}(a, b, b) = a$ and $t_{(a,b,c,d)}(c, c, d) = d$ if and only if

$$(a, d) \in \mathrm{Sg}^{\mathbb{A}^2}(\{(a, c), (b, c), (b, d)\}.$$

- Since subalgebra generation is linear (sort of) in the size of the algebra, then the above leads to an algorithm for testing for a Maltsev term whose runtime can be bounded by a polynomial of degree 6 in the size of the algebra.

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*
- *[Val. Willard] generating a congruence k-permutable variety.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*
- *[Val. Willard] generating a congruence k-permutable variety.*
- *[Barto, Kozik] having a k-ary cyclic term.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*
- *[Val. Willard] generating a congruence k-permutable variety.*
- *[Barto, Kozik] having a k-ary cyclic term.*
- *[Kazda] having a k-ary weak near unanimity term.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*
- *[Val. Willard] generating a congruence k-permutable variety.*
- *[Barto, Kozik] having a k-ary cyclic term.*
- *[Kazda] having a k-ary weak near unanimity term.*
- *[Val.] having a sequence of Jónsson terms of length 4.*

More local term results

- A number of other well known strong idempotent Maltsev conditions can also be characterized via suitable local term conditions, for finite idempotent algebras.

Theorem

For a finite idempotent algebra $\mathbb{A}$, the following properties can be characterized using local terms, and hence can be tested in polynomial-time: for a fixed value of k,

- *[Horowitz] having a k-ary near unanimity term.*
- *[Horowitz] having a k-edge term.*
- *[Val. Willard] generating a congruence k-permutable variety.*
- *[Barto, Kozik] having a k-ary cyclic term.*
- *[Kazda] having a k-ary weak near unanimity term.*
- *[Val.] having a sequence of Jónsson terms of length 4.*
- *[DeMeo, Freese, Val.] having a difference term.*

Example: Testing for a 4-ary near unanimity term

- Earlier we saw an algorithm to decide if a finite algebra $\mathbb{A}$ has a 4-ary near unanimity term.

Example: Testing for a 4-ary near unanimity term

- Earlier we saw an algorithm to decide if a finite algebra $\mathbb{A}$ has a 4-ary near unanimity term.
- Its runtime is exponential in the size of $\mathbb{A}$, since it needed to first build the free algebra on 2 generators in $\mathbf{V}(\mathbb{A})$

Example: Testing for a 4-ary near unanimity term

- Earlier we saw an algorithm to decide if a finite algebra $\mathbb{A}$ has a 4-ary near unanimity term.
- Its runtime is exponential in the size of $\mathbb{A}$, since it needed to first build the free algebra on 2 generators in $\mathbf{V}(\mathbb{A})$
- When $\mathbb{A}$ is idempotent, then we only need to perform the following test: for all x and y in $\mathbb{A}$, the tuple (x, x, x, x) is in the subalgebra of $\mathbb{A}^4$ generated by

$$\{(y, x, x, x), (x, y, x, x), (x, x, y, x), (x, x, x, y)\}$$

Example: Testing for a 4-ary near unanimity term

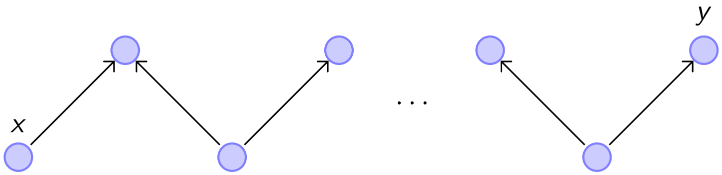
- Earlier we saw an algorithm to decide if a finite algebra $\mathbb{A}$ has a 4-ary near unanimity term.
- Its runtime is exponential in the size of $\mathbb{A}$, since it needed to first build the free algebra on 2 generators in $\mathbf{V}(\mathbb{A})$
- When $\mathbb{A}$ is idempotent, then we only need to perform the following test: for all x and y in $\mathbb{A}$, the tuple (x, x, x, x) is in the subalgebra of $\mathbb{A}^4$ generated by

$$\{(y, x, x, x), (x, y, x, x), (x, x, y, x), (x, x, x, y)\}$$

- Let's try this on our demo algebra.

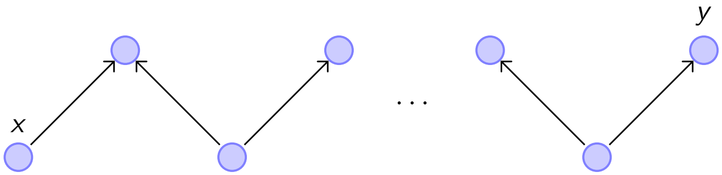
A general result

- Alex Kazda and I have been able to unify several of the results and arguments from the previous theorem to show that a fairly large class of idempotent Maltsev conditions can be decided in polynomial-time for finite idempotent algebras.



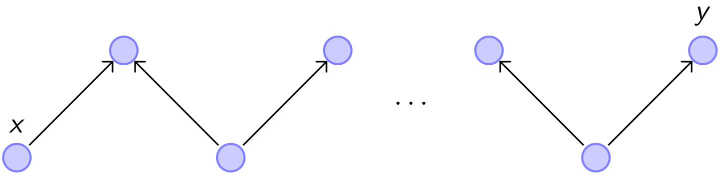
A general result

- Alex Kazda and I have been able to unify several of the results and arguments from the previous theorem to show that a fairly large class of idempotent Maltsev conditions can be decided in polynomial-time for finite idempotent algebras.
- These Maltsev conditions can all be specified using a certain type of labelled path.



A general result

- Alex Kazda and I have been able to unify several of the results and arguments from the previous theorem to show that a fairly large class of idempotent Maltsev conditions can be decided in polynomial-time for finite idempotent algebras.
- These Maltsev conditions can all be specified using a certain type of labelled path.
- For example, the following path captures the property of having a sequence of Jónsson terms of length equal to the length of the path.



A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ *can be decided by polynomial-time algorithms: for a fixed value of* k*,*

- $\mathbb{A}$ *has a sequence of* k *(directed) Jónsson terms,*

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

- *$\mathbb{A}$ has a sequence of k (directed) Jónsson terms,*
- *$\mathbb{A}$ has a sequence of k (directed) Gumm terms,*

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

- $\mathbb{A}$ *has a sequence of k (directed) Jónsson terms,*
- $\mathbb{A}$ *has a sequence of k (directed) Gumm terms,*
- $\mathbb{A}$ *has a sequence of k Hagemann-Mitschke terms (for congruence $(k+1)$-permutability).*

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

- *$\mathbb{A}$ has a sequence of k (directed) Jónsson terms,*
- *$\mathbb{A}$ has a sequence of k (directed) Gumm terms,*
- *$\mathbb{A}$ has a sequence of k Hagemann-Mitschke terms (for congruence $(k+1)$-permutability).*

Remark

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

- *$\mathbb{A}$ has a sequence of k (directed) Jónsson terms,*
- *$\mathbb{A}$ has a sequence of k (directed) Gumm terms,*
- *$\mathbb{A}$ has a sequence of k Hagemann-Mitschke terms (for congruence $(k+1)$-permutability).*

Remark

- *The polynomial-time algorithms that we produce are based on the "local terms" method.*

A general result

Theorem (Kazda, Val.)

The following strong Maltsev conditions are "path definable" and so for a finite idempotent algebra $\mathbb{A}$ can be decided by polynomial-time algorithms: for a fixed value of k,

- *$\mathbb{A}$ has a sequence of k (directed) Jónsson terms,*
- *$\mathbb{A}$ has a sequence of k (directed) Gumm terms,*
- *$\mathbb{A}$ has a sequence of k Hagemann-Mitschke terms (for congruence $(k+1)$-permutability).*

Remark

- *The polynomial-time algorithms that we produce are based on the "local terms" method.*
- *We show that a path definable Maltsev condition will hold for a finite idempotent algebra if and only if the algebra has enough local term versions of the Maltsev condition.*

A conjecture

Observation

A conjecture

Observation

- All of these Maltsev conditions, and the ones from the previous theorem are strong, idempotent, and linear.

A conjecture

Observation

- All of these Maltsev conditions, and the ones from the previous theorem are strong, idempotent, and linear.
- A Maltsev condition is linear if its equations do not involve compositions of any of its operations.

A conjecture

Observation

- All of these Maltsev conditions, and the ones from the previous theorem are strong, idempotent, and linear.
- A Maltsev condition is linear if its equations do not involve compositions of any of its operations.

Conjecture (Kazda, Val.)

If Σ is a strong, idempotent, linear Maltsev condition, then there is a polynomial-time algorithm to decide if a given finite idempotent algebra satisfies it.

The curious case of the minority term

Remarks

The curious case of the minority term

Remarks

- *A* *minority term* *is a ternary term* $m(x, y, z)$ *that satisfies:*

$$m(x, y, y) \approx m(y, x, y) \approx m(y, y, x) \approx x.$$

The curious case of the minority term

Remarks

- *A minority term is a ternary term* $m(x, y, z)$ *that satisfies:*

$$m(x, y, y) \approx m(y, x, y) \approx m(y, y, x) \approx x.$$

- *The associated Maltsev condition is not path definable and so is not covered by the previous theorem.*

The curious case of the minority term

Remarks

- *A minority term is a ternary term $m(x, y, z)$ that satisfies:*

$$m(x, y, y) \approx m(y, x, y) \approx m(y, y, x) \approx x.$$

- *The associated Maltsev condition is not path definable and so is not covered by the previous theorem.*
- *Dmitry Zhuk has shown that there cannot be a local-term based polynomial-time algorithm to decide this condition for idempotent algebras.*

The curious case of the minority term

Remarks

- *A minority term is a ternary term $m(x, y, z)$ that satisfies:*

$$m(x, y, y) \approx m(y, x, y) \approx m(y, y, x) \approx x.$$

- *The associated Maltsev condition is not path definable and so is not covered by the previous theorem.*
- *Dmitry Zhuk has shown that there cannot be a local-term based polynomial-time algorithm to decide this condition for idempotent algebras.*
- *At present it is currently unknown if there is any type of polynomial-time algorithm to decide this condition, but*

The curious case of the minority term

Remarks

- *A minority term is a ternary term $m(x, y, z)$ that satisfies:*

$$m(x, y, y) \approx m(y, x, y) \approx m(y, y, x) \approx x.$$

- *The associated Maltsev condition is not path definable and so is not covered by the previous theorem.*
- *Dmitry Zhuk has shown that there cannot be a local-term based polynomial-time algorithm to decide this condition for idempotent algebras.*
- *At present it is currently unknown if there is any type of polynomial-time algorithm to decide this condition, but*
- *Kazda, Oprsal and I have shown that this decision problem is in the class* **NP**.

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.
- This raises the question of whether there might be some strong, idempotent, essentially non-linear Maltsev conditions that can also be decided, for idempotent algebras, in polynomial-time.

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.
- This raises the question of whether there might be some strong, idempotent, essentially non-linear Maltsev conditions that can also be decided, for idempotent algebras, in polynomial-time.
- A natural test-case is that of having a semilattice term.

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.
- This raises the question of whether there might be some strong, idempotent, essentially non-linear Maltsev conditions that can also be decided, for idempotent algebras, in polynomial-time.
- A natural test-case is that of having a semilattice term.
- This strong idempotent Maltsev condition is not equivalent to a linear one, but is rather simple.

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.
- This raises the question of whether there might be some strong, idempotent, essentially non-linear Maltsev conditions that can also be decided, for idempotent algebras, in polynomial-time.
- A natural test-case is that of having a semilattice term.
- This strong idempotent Maltsev condition is not equivalent to a linear one, but is rather simple.
- Ralph Freese and I have shown that testing for this condition, in general, is an EXP-TIME complete problem, and ...

Beyond linear Maltsev conditions

- The conjecture and all of the positive results mentioned so far involve linear Maltsev conditions.
- This raises the question of whether there might be some strong, idempotent, essentially non-linear Maltsev conditions that can also be decided, for idempotent algebras, in polynomial-time.
- A natural test-case is that of having a semilattice term.
- This strong idempotent Maltsev condition is not equivalent to a linear one, but is rather simple.
- Ralph Freese and I have shown that testing for this condition, in general, is an EXP-TIME complete problem, and ...

Theorem (Freese, Nation, Val.)

Testing for a semilattice term, even for finite idempotent algebras, is an EXP-TIME complete problem.

Conclusion

Open problems

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?
- What is the complexity of testing for a 2-semilattice term (a commutative idempotent binary term $x \wedge y$ that satisfies: $x \wedge (x \wedge y) \approx x \wedge y$)?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?
- What is the complexity of testing for a 2-semilattice term (a commutative idempotent binary term $x \wedge y$ that satisfies: $x \wedge (x \wedge y) \approx x \wedge y$)?
- Is there a strong, idempotent, non-linear Maltsev condition that can be tested in polynomial-time (or by a non-deterministic poly-time algorithm), for idempotent algebras?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?
- What is the complexity of testing for a 2-semilattice term (a commutative idempotent binary term $x \wedge y$ that satisfies: $x \wedge (x \wedge y) \approx x \wedge y$)?
- Is there a strong, idempotent, non-linear Maltsev condition that can be tested in polynomial-time (or by a non-deterministic poly-time algorithm), for idempotent algebras?
- Is there a non-trivial, strong idempotent linear Maltsev condition that can be tested, for general finite algebras, in less than EXP-TIME?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?
- What is the complexity of testing for a 2-semilattice term (a commutative idempotent binary term $x \wedge y$ that satisfies: $x \wedge (x \wedge y) \approx x \wedge y$)?
- Is there a strong, idempotent, non-linear Maltsev condition that can be tested in polynomial-time (or by a non-deterministic poly-time algorithm), for idempotent algebras?
- Is there a non-trivial, strong idempotent linear Maltsev condition that can be tested, for general finite algebras, in less than EXP-TIME?
- What is the complexity of testing for a Maltsev term, or a majority term, for arbitrary finite algebras?

Conclusion

Open problems

- What is the complexity of testing for a minority term, in the idempotent case (it is in **NP**)?
- Is the strong, idempotent, linear Maltsev conjecture true?
- What is the complexity of testing for a 2-semilattice term (a commutative idempotent binary term $x \wedge y$ that satisfies: $x \wedge (x \wedge y) \approx x \wedge y$)?
- Is there a strong, idempotent, non-linear Maltsev condition that can be tested in polynomial-time (or by a non-deterministic poly-time algorithm), for idempotent algebras?
- Is there a non-trivial, strong idempotent linear Maltsev condition that can be tested, for general finite algebras, in less than EXP-TIME?
- What is the complexity of testing for a Maltsev term, or a majority term, for arbitrary finite algebras?
- What is the complexity of deciding if a finite algebra is primal?