Algorithmic problems for equations and algebras

George F. McNulty

Department of Mathematics
University of South Carolina

June 2007
International Conference on Order, Algebra, and Logics
Vanderbilt University

Or How Hard Must It Be?

Outline

Outline

Outline

Outline

Outline

A Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

A Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

A Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

How hard must that be?

The Components of the Problem

1. The problem asks for a method or algorithm. We take this as a request for a computer program.
2. The method (i.e. the computer program) has an input, in this case a given equation.
3. The computer program is suppose to determine whether the input has a certain property, in this case whether it is true in some nontrivial lattice.
4. How hard must that be?

The Components of the Problem

1. The problem asks for a method or algorithm. We take this as a request for a computer program.
2. The method (i.e. the computer program) has an input, in this case a given equation.
3. The computer program is suppose to determine whether the input has a certain property, in this case whether it is true in some nontrivial lattice.
4. How hard must that be?

The Components of the Problem

1. The problem asks for a method or algorithm. We take this as a request for a computer program.
2. The method (i.e. the computer program) has an input, in this case a given equation.
3. The computer program is suppose to determine whether the input has a certain property, in this case whether it is true in some nontrivial lattice.
4. How hard must that be?

The Components of the Problem

1. The problem asks for a method or algorithm. We take this as a request for a computer program.
2. The method (i.e. the computer program) has an input, in this case a given equation.
3. The computer program is suppose to determine whether the input has a certain property, in this case whether it is true in some nontrivial lattice.
4. How hard must that be?

Inputs

Roughly speaking, anything which can be typed into a computer keyboard using only finitely many keystrokes is a reasonable input.

We will use $\mathcal{I}$ to denote the set of all such inputs.

Of course, computer programs can be entered via computer keyboards. We use $\mathcal{M}$ to denote the set of all programs.

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), ($.

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), ($.

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), (.$

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), (.$

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), ($.

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), ($.

That is, our inputs are to be equations in the signature of lattices.

Inputs

However, in framing our problems, we will make more particular specifications about the inputs.

In the case of our sample problem these inputs are to be equations built up using

- the operation symbols $\wedge$ and $\vee$ of lattice theory,
- a countably infinite list $x_0, x_1, x_2, \ldots$ of variables,
- a symbol $\approx$ for equality, and
- punctuation $), ($.

That is, our inputs are to be equations in the signature of lattices.

Following the usage in the computer science community, we render this part of our sample problem as:

INSTANCE: An equation $s \approx t$ in the signature of lattices.

Properties

While the inputs associated with a problem like our sample problem are severely restricted, the properties are given freer range. Suppose that $\mathcal{S}$ is the set of all inputs. (In our example, $\mathcal{S}$ is the set of all equations in the signature of lattices.) We allow the property associated with one of our problems to be any subset $\mathcal{P} \subseteq \mathcal{S}$.

Properties

While the inputs associated with a problem like our sample problem are severely restricted, the properties are given freer range. Suppose that $\mathcal{S}$ is the set of all inputs. (In our example, $\mathcal{S}$ is the set of all equations in the signature of lattices.) We allow the property associated with one of our problems to be any subset $\mathcal{P} \subseteq \mathcal{S}$.

The problems we consider have the form

INSTANCE: $\sigma \in \mathcal{S}$.

QUESTION: Is $\sigma \in \mathcal{P}$*?*

Properties

While the inputs associated with a problem like our sample problem are severely restricted, the properties are given freer range. Suppose that $\mathcal{S}$ is the set of all inputs. (In our example, $\mathcal{S}$ is the set of all equations in the signature of lattices.) We allow the property associated with one of our problems to be any subset $\mathcal{P} \subseteq \mathcal{S}$.

The problems we consider have the form

INSTANCE: $\sigma \in \mathcal{S}$.

QUESTION: Is $\sigma \in \mathcal{P}$*?*

So our problems are completely determined by specifying $\mathcal{S}$ and one of its subsets $\mathcal{P}$. We will refer to $(\mathcal{S}, \mathcal{P})$ as a computational problem.

Our Sample Problem, Reprise

INSTANCE: An equation $s \approx t$ in the signature of lattices.

QUESTION: Is $s \approx t$ true in some nontrivial lattice?

Complexity of Inputs

In these tutorials, we will only be concerned with inputs that are equations

Complexity of Inputs

In these tutorials, we will only be concerned with inputs that are equations (or finite sets of equations or pairs of equations or . . .)

Complexity of Inputs

In these tutorials, we will only be concerned with inputs that are equations (or finite sets of equations or pairs of equations or . . .) or finite algebras

Complexity of Inputs

In these tutorials, we will only be concerned with inputs that are equations (or finite sets of equations or pairs of equations or . . .) or finite algebras (or finite sets of finite algebras or pairs of finite algebras . . .).

Complexity of Inputs

For the cases when $\mathcal{S}$ arises in this way (and many other cases) there are three important features:

1. There is a computationally cheap way to scan a sequence of keystrokes to determine if it belongs to $\mathcal{S}$;
2. There is at least one obvious way to assign a natural number to each member of $\mathcal{S}$ that can be understood as its complexity. We will use $\|\sigma\|$ to denote the obvious complexity of σ.
3. For each n there are only finitely many inputs of complexity n.

Complexity of Inputs

For the cases when $\mathcal{S}$ arises in this way (and many other cases) there are three important features:

1. There is a computationally cheap way to scan a sequence of keystrokes to determine if it belongs to $\mathcal{S}$;
2. There is at least one obvious way to assign a natural number to each member of $\mathcal{S}$ that can be understood as its complexity. We will use $\|\sigma\|$ to denote the obvious complexity of σ.
3. For each n there are only finitely many inputs of complexity n.

Complexity of Inputs

For the cases when $\mathcal{S}$ arises in this way (and many other cases) there are three important features:

1. There is a computationally cheap way to scan a sequence of keystrokes to determine if it belongs to $\mathcal{S}$;
2. There is at least one obvious way to assign a natural number to each member of $\mathcal{S}$ that can be understood as its complexity. We will use $\|\sigma\|$ to denote the obvious complexity of σ.
3. For each n there are only finitely many inputs of complexity n.

Complexity of Inputs

For the cases when $\mathcal{S}$ arises in this way (and many other cases) there are three important features:

1. There is a computationally cheap way to scan a sequence of keystrokes to determine if it belongs to $\mathcal{S}$;
2. There is at least one obvious way to assign a natural number to each member of $\mathcal{S}$ that can be understood as its complexity. We will use $\|\sigma\|$ to denote the obvious complexity of σ.
3. For each n there are only finitely many inputs of complexity n.

Complexity of Inputs

For the cases when $\mathcal{S}$ arises in this way (and many other cases) there are three important features:

1. There is a computationally cheap way to scan a sequence of keystrokes to determine if it belongs to $\mathcal{S}$;
2. There is at least one obvious way to assign a natural number to each member of $\mathcal{S}$ that can be understood as its complexity. We will use $\|\sigma\|$ to denote the obvious complexity of σ.
3. For each n there are only finitely many inputs of complexity n.

Loosely speaking, $\|\sigma\|$ is intended to measure the compuational resources it takes for the computer to absorb the input σ before actually doing anything interesting with it.

How Hard Must It Be?

It might be impossible.

How Hard Must It Be?

It might be impossible. There might not be any algorithm which does the job. In this case, we say that the problem is **algorithmically unsolvable** or that $\mathcal{P}$ is **undecidable**.

How Hard Must It Be?

If there is at least one computer program that does the job, then such a program will provide a function which is an **upper** bound on the amount of time and space the execution of this program requires, given any input of complexity n.

How Hard Must It Be?

If there is at least one computer program that does the job, then such a program will provide a function which is an **upper** bound on the amount of time and space the execution of this program requires, given any input of complexity n. The art of the upper bound requires to insightful creation of clever algorithms and the detailed understanding of their operation.

How Hard Must It Be?

The art of the lower bound, of answering the question

"How hard must it be?"

is another thing.

How Hard Must It Be, Really?

Once there is one algorithm to solve the problem, there will certainly many others. Some will be less demanding than others on our computational resources. To answer the question in the title above we must have a lower bound on the complexity of all these algorithms.

How Hard Must It Be, Really?

To conclude, for example, that a problem cannot be done in polynomial time apparently requires us to consider any algorithm M which answers our problem and to show that for every polynomial $p(x)$ and for arbitrarily large n there is $\sigma \in \mathcal{S}$ with $\|\sigma\| = n$ so that M requires more time than $p(n)$ to determine whether $\sigma \in \mathcal{P}$.

How Hard Must It Be, Really?

Evidently, answering the question

"How hard must it be?"

can be, well .

How Hard Must It Be, Really?

Evidently, answering the question

"How hard must it be?"

can be, well hard.

A Lot Is Already Known

These kinds of questions belong to computational complexity, one of the central subjects pursued vigorously by our colleagues in computer science for the past forty years,

A Lot Is Already Known

These kinds of questions belong to the subject of decidability and undecidability, which has attracted the attention of mathematical logicians for the past eighty years,

A Lot Is Already Known

These kinds of questions belong to the mathematician's fascination with algorithms since before the ancients found out how to bisect angles with straightedge and compass.

A Lot Is Already Known

So quite a lot is known at this late date.

A Lot Is Already Known

We have all heard at least rumors of a hierarchy of complexity classes **LOGSPACE, P, NP, co-NP, EXPTIME, PSPACE,**....

While much is known about this hierarchy, there are some great challenges that remain outstanding.

Also the placing of many particular problems, chiefly of a combinatorial nature, within this hierarchy has already been achieved.

A Lot Is Already Known

Happily, we can take advantage of this accumulated knowledge to address problems concerning equations and finite algbras.

- There will be an interesting collection of talks on recent results of this kind here at our conference.
- Also the Workshop on the Constraint Satisfaction Problem immediately following our conference continues this theme.
- And in July there will be a conference in Szeged focussed largely on these matters.

A Lot Is Already Known

Happily, we can take advantage of this accumulated knowledge to address problems concerning equations and finite algbras.

- There will be an interesting collection of talks on recent results of this kind here at our conference.
- Also the Workshop on the Constraint Satisfaction Problem immediately following our conference continues this theme.
- And in July there will be a conference in Szeged focussed largely on these matters.

A Lot Is Already Known

Happily, we can take advantage of this accumulated knowledge to address problems concerning equations and finite algbras.

- There will be an interesting collection of talks on recent results of this kind here at our conference.
- Also the Workshop on the Constraint Satisfaction Problem immediately following our conference continues this theme.
- And in July there will be a conference in Szeged focussed largely on these matters.

A Lot Is Already Known

Happily, we can take advantage of this accumulated knowledge to address problems concerning equations and finite algbras.

- There will be an interesting collection of talks on recent results of this kind here at our conference.
- Also the Workshop on the Constraint Satisfaction Problem immediately following our conference continues this theme.
- And in July there will be a conference in Szeged focussed largely on these matters.

Exponential Time Complexity

Given a set $\mathcal{S}$ of inputs and a subset $\mathcal{P} \subseteq \mathcal{S}$ we will say that the associated problem $(\mathcal{S}, \mathcal{P})$ belongs to **EXPTIME** provided there is a polynomial $p(n)$ and a computer program M for solving the problem such that for all $\sigma \in \mathcal{S}$ the program M determines whether $\sigma \in \mathcal{P}$ after taking no more than $2^{p(\|\sigma\|)}$ steps.

Exponential Time Complexity

The problem associated with $\mathcal{S}$ and $\mathcal{P}$ is said to be EXPTIME **difficult** provided there is a real number $c > 0$ so that for any computer program M which solves the problem and for arbitrarily large values of n there is $\sigma \in \mathcal{S}$ with $\|\sigma\| = n$ so that the computation of M on input σ runs for at least 2^{cn} steps.

Outline

Polynomial Time Turing Reduction of One Problem to Another

Let $(\mathcal{S}, \mathcal{P})$ be a computational problem. The capacity of a computer programming system can be enhanced by allowing it the use of an *oracle* for $(\mathcal{S}, \mathcal{P})$: in the course of a computation the oracle, at the cost of a single step, will provide the correct answer to questions of the form "Is $\sigma \in \mathcal{P}$?" Of course, the time and space needed to construct any particular σ is charged to the resources used by the computation.

Polynomial Time Turing Reduction of One Problem to Another

Let $(\mathcal{S}_0, \mathcal{P}_0)$ and $(\mathcal{S}_1, \mathcal{P}_1)$ be two computational problems. We say that $(\mathcal{S}_1, \mathcal{P}_1)$ is **polynomially Turing reducible** to $(\mathcal{S}_0, \mathcal{P}_0)$ provided there is a polynomial $p(n)$ and a program M with an oracle for $(\mathcal{S}_0, \mathcal{P}_0)$ so that M decides $\sigma \in \mathcal{P}_1$ in no more than $p(\|\sigma\|)$ steps, for all $\sigma \in \mathcal{S}_1$.

Many-One Reductions of One Problem to Another

Let $(\mathcal{S}_0, \mathcal{P}_0)$ and $(\mathcal{S}_1, \mathcal{P}_1)$ be two computational problems and let $\Phi : \mathcal{S}_1 \to \mathcal{S}_0$. We say that Φ **reduces** $(\mathcal{S}_1, \mathcal{P}_1)$ to $(\mathcal{S}_0, \mathcal{P}_0)$ provided

$$\sigma \in \mathcal{P}_1 \text{ if and only if } \Phi(\sigma) \in \mathcal{P}_0$$

for all $\sigma \in \mathcal{S}_1$.

Many-One Reductions of One Problem to Another

This reduction is **polynomial time** if the there is a polynomial $p(n)$ so that the computation of $\Phi(\sigma)$ concludes after no more than $p(\|\sigma\|)$ steps for all σ. It is **logspace** if the computation of $\Phi(\sigma)$ requires no more than roughly $\log(\|\sigma\|)$ space in memory (the space occupied by the input and the output held unavailable for computation).

Many-One Reductions of One Problem to Another

Suppose that M is a computer program that resolves $(\mathcal{S}_0, \mathcal{P}_0)$.

Make a new program M^* to resolve the problem $(\mathcal{S}_1, \mathcal{P}_1)$. Here is what M^* does with input $\sigma \in \mathcal{S}_1$:

1. Construct $\Phi(\sigma)$.
2. Launch M on $\Phi(\sigma)$ to determine whether $\Phi(\sigma) \in \mathcal{P}_0$
3. Return the result.

Many-One Reductions of One Problem to Another

Suppose that M is a computer program that resolves $(\mathcal{S}_0, \mathcal{P}_0)$.

Make a new program M^* to resolve the problem $(\mathcal{S}_1, \mathcal{P}_1)$. Here is what M^* does with input $\sigma \in \mathcal{S}_1$:

1. Construct $\Phi(\sigma)$.
2. Launch M on $\Phi(\sigma)$ to determine whether $\Phi(\sigma) \in \mathcal{P}_0$
3. Return the result.

Many-One Reductions of One Problem to Another

Suppose that M is a computer program that resolves $(\mathcal{S}_0, \mathcal{P}_0)$.

Make a new program M^* to resolve the problem $(\mathcal{S}_1, \mathcal{P}_1)$. Here is what M^* does with input $\sigma \in \mathcal{S}_1$:

1. Construct $\Phi(\sigma)$.
2. Launch M on $\Phi(\sigma)$ to determine whether $\Phi(\sigma) \in \mathcal{P}_0$
3. Return the result.

Ensuring that $(\mathcal{S}_1, \mathcal{P}_1)$ is easy enough

Suppose that $f(n)$ is a function giving an upper bound to the number of steps M takes on inputs of complexity n.
The number of steps M^* takes on input σ is no more than $t + f(\|\Phi(\sigma)\|)$, where t is the number of steps needed to construct $\Phi(\sigma)$ from σ.

Consequently, if Φ is cheap to compute and $\Phi(\sigma)$ is not very much more complex than σ, then M^* will resolve $(\mathcal{S}_1, \mathcal{P}_1)$ in not so many more steps than it takes M to resolve $(\mathcal{S}_0, \mathcal{P}_0)$.

Ensuring that $(\mathcal{S}_0, \mathcal{P}_0)$ is hard enough

On the other hand, suppose you have a lower bound $g(n)$ on the time complexity of $(\mathcal{S}_1, \mathcal{P}_1)$. Then for inputs σ of arbitrarily large complexity we know that M^* must take at least $g(\|\sigma\|)$ steps. For such σ, we are forced to conclude that M takes at least $g(\|\sigma\|) - t$ steps upon input $\Phi(\sigma)$, where t is again the number of steps needed to compute $\Phi(\sigma)$ from σ.

Consequently, if Φ is cheap to compute and $\Phi(\sigma)$ is not very much more complex than σ, then $(\mathcal{S}_0, \mathcal{P}_0)$ cannot be much easier to resolve than $(\mathcal{S}_1, \mathcal{P}_1)$.

Comparing Reductions

$(\mathcal{S}_1, \mathcal{P}_1)$ is logspace reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

$\Downarrow$

$(\mathcal{S}_1, \mathcal{P}_1)$ is polynomial time reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

Comparing Reductions

$(\mathcal{S}_1, \mathcal{P}_1)$ is logspace reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

$\Downarrow$

$(\mathcal{S}_1, \mathcal{P}_1)$ is polynomial time reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

$\Downarrow$

$(\mathcal{S}_1, \mathcal{P}_1)$ is polynomially Turing reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

Comparing Reductions

$(\mathcal{S}_1, \mathcal{P}_1)$ is logspace reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

$\Downarrow$

$(\mathcal{S}_1, \mathcal{P}_1)$ is polynomial time reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

$\Downarrow$

$(\mathcal{S}_1, \mathcal{P}_1)$ is polynomially Turing reducible to $(\mathcal{S}_0, \mathcal{P}_0)$

Of these three reducibilities, the logspace many-one reduction is the most desirable because it provides the tightest relationship between the complexities of the two problems. These three are not the only reducibilities.

Hard Problems

Let C be a complexity class (like EXPTIME or NP). A problem $(\mathcal{S}, \mathcal{P})$ is said to be C **hard** provided every problem in C is reducible to $(\mathcal{S}, \mathcal{P})$. A problem is C **complete** if it is C hard and also belongs to C.

Hard Problems

Let C be a complexity class (like EXPTIME or NP). A problem $(\mathcal{S}, \mathcal{P})$ is said to be C **hard** provided every problem in C is reducible to $(\mathcal{S}, \mathcal{P})$. A problem is C **complete** if it is C hard and also belongs to C.

Each notion of reducible gives rise to its associated notions of hardness and completeness.

Hard Problems

Let C be a complexity class (like EXPTIME or NP). A problem $(\mathcal{S}, \mathcal{P})$ is said to be C **hard** provided every problem in C is reducible to $(\mathcal{S}, \mathcal{P})$. A problem is C **complete** if it is C hard and also belongs to C.

Each notion of reducible gives rise to its associated notions of hardness and completeness.

The C complete problems might be regarded as the most demanding problems in C. One should take this with a grain of salt. The density of difficult instances might be low for a complete problem, while an incomplete problem might on the whole be more troublesome. Rather it is more to the point to say that a computer program for a complete problem can be adapted to solve any other problem in C so that running the adapted program is only a bit more costly than the original.

A Way to Proceed

1. Find in the literature a problem $(\mathcal{S}, \mathcal{P})$ with a known complexity.
2. Cleverly invent a map Φ and prove that it gives the desired reduction relation between $(\mathcal{S}, \mathcal{P})$ and the problem in which you are interested, and
3. Prove Φ is cheap enough to compute and does not make $\Phi(\sigma)$ much more complex than σ.

A Way to Proceed

1. Find in the literature a problem $(\mathcal{S}, \mathcal{P})$ with a known complexity.
2. Cleverly invent a map Φ and prove that it gives the desired reduction relation between $(\mathcal{S}, \mathcal{P})$ and the problem in which you are interested, and
3. Prove Φ is cheap enough to compute and does not make $\Phi(\sigma)$ much more complex than σ.

A Way to Proceed

1. Find in the literature a problem $(\mathcal{S}, \mathcal{P})$ with a known complexity.
2. Cleverly invent a map Φ and prove that it gives the desired reduction relation between $(\mathcal{S}, \mathcal{P})$ and the problem in which you are interested, and
3. Prove Φ is cheap enough to compute and does not make $\Phi(\sigma)$ much more complex than σ.

Outline

Fast Growing Functions

Explosive Functions

Call a function $f : \mathbb{R}^+ \to \mathbb{R}^+$ **explosive** if f is strictly increasing and

$$\lim_{n\to\infty} \frac{q(n)}{f(n)} = 0 \text{ and}$$
$$\lim_{n\to\infty} \frac{f(\alpha n + \beta)}{f(n)} = 0$$

whenever $q(n)$ is a polynomial, $0 \leq \alpha < 1$, and $0 \leq \beta$.

2^n and 2^{2^n} are examples of explosive functions.

Simulation and Diagonalization

An Explosive Lower Bound

Let $\mathcal{S}$ be a set of inputs with $\mathcal{P} \subseteq \mathcal{S}$. Let f be an explosive function, $p(n)$ be a polynomial, and d be a nonnegative real. Let $\Psi : \mathcal{M} \times \mathcal{I} \to \mathcal{S}$. If for all $M \in \mathcal{M}$ and all $x \in \mathcal{I}$

1. $\|\Psi(M, x)\| \leq d \cdot (\|M\| + \|x\|)$;
2. $\Psi(M, x)$ can be computed from M and x in time bounded by $p(\|M\| + \|x\|)$;
3. $\Psi(M, x) \in \mathcal{P}$ if and only if M on input x computes for more than $f(n)$ steps,

then there is $c > 0$ so that deciding $\mathcal{P}$ is at least $f(cn)$ difficult.

Simulation and Diagonalization

An Explosive Lower Bound

Let $\mathcal{S}$ be a set of inputs with $\mathcal{P} \subseteq \mathcal{S}$. Let f be an explosive function, $p(n)$ be a polynomial, and d be a nonnegative real. Let $\Psi : \mathcal{M} \times \mathcal{I} \to \mathcal{S}$. If for all $M \in \mathcal{M}$ and all $x \in \mathcal{I}$

1. $\|\Psi(M, x)\| \leq d \cdot (\|M\| + \|x\|)$;
2. $\Psi(M, x)$ can be computed from M and x in time bounded by $p(\|M\| + \|x\|)$;
3. $\Psi(M, x) \in \mathcal{P}$ if and only if M on input x computes for more than $f(n)$ steps,

then there is $c > 0$ so that deciding $\mathcal{P}$ is at least $f(cn)$ difficult.

Simulation and Diagonalization

An Explosive Lower Bound

Let $\mathcal{S}$ be a set of inputs with $\mathcal{P} \subseteq \mathcal{S}$. Let f be an explosive function, $p(n)$ be a polynomial, and d be a nonnegative real. Let $\Psi : \mathcal{M} \times \mathcal{I} \to \mathcal{S}$. If for all $M \in \mathcal{M}$ and all $x \in \mathcal{I}$

1. $\|\Psi(M, x)\| \leq d \cdot (\|M\| + \|x\|)$;
2. $\Psi(M, x)$ can be computed from M and x in time bounded by $p(\|M\| + \|x\|)$;
3. $\Psi(M, x) \in \mathcal{P}$ if and only if M on input x computes for more than $f(n)$ steps,

then there is $c > 0$ so that deciding $\mathcal{P}$ is at least $f(cn)$ difficult.

Simulation and Diagonalization

An Explosive Lower Bound

Let $\mathcal{S}$ be a set of inputs with $\mathcal{P} \subseteq \mathcal{S}$. Let f be an explosive function, $p(n)$ be a polynomial, and d be a nonnegative real. Let $\Psi : \mathcal{M} \times \mathcal{I} \to \mathcal{S}$. If for all $M \in \mathcal{M}$ and all $x \in \mathcal{I}$

1. $\|\Psi(M, x)\| \leq d \cdot (\|M\| + \|x\|)$;
2. $\Psi(M, x)$ can be computed from M and x in time bounded by $p(\|M\| + \|x\|)$;
3. $\Psi(M, x) \in \mathcal{P}$ if and only if M on input x computes for more than $f(n)$ steps,

then there is $c > 0$ so that deciding $\mathcal{P}$ is at least $f(cn)$ difficult.

A Sketch

Let M be any program for deciding $\mathcal{P}$. Devise a new program M^* which does the following on input x:

- If x is a program, then M^* first constructs $\Psi(x, x)$ and then launches M on $\Psi(x, x)$. If M determines that $\Psi(x, x)$ has $\mathcal{P}$, then M^* halts. If M determines that $\Psi(x, x)$ does not have $\mathcal{P}$ then M^* goes into an infinite loop.
- If x is not a program, then M^* goes into an infinite loop.

A Sketch

Let M be any program for deciding $\mathcal{P}$. Devise a new program M^* which does the following on input x:

- If x is a program, then M^* first constructs $\Psi(x, x)$ and then launches M on $\Psi(x, x)$. If M determines that $\Psi(x, x)$ has $\mathcal{P}$, then M^* halts. If M determines that $\Psi(x, x)$ does not have $\mathcal{P}$ then M^* goes into an infinite loop.
- If x is not a program, then M^* goes into an infinite loop.

Suppose that x is a program. Then we have

$$\begin{aligned} M^* \text{ halts on } x &\Leftrightarrow \Psi(x, x) \in \mathcal{P} \\ &\Leftrightarrow x \text{ does not halt on } x \text{ within } f(\|x\|) \text{ steps.} \end{aligned}$$

More Sketch

In particular, M^* halts on M^* if and only if M^* does not halt on M^* within $f(\|M^*\|)$ steps. Hence

$\Psi(M^*, M^*)$ has property $\mathcal{P}$

Still More Sketch

How long does the computation of M^* upon input M^* take?

1. First $\Psi(M^*, M^*)$ is built. This takes no more than $p(2\|M^*\|) = p(\|M^*\| + \|M^*\|)$ steps.
2. M determines that $\Psi(M^*, M^*)$ has property $\mathcal{P}$. Let us say this takes t steps.
3. Then M^* halts.

$p(2\|M^*\|) + t > f(\|M^*\|)$

Our Sketch Ends

It does no harm to suppose that $d > 1$. We take

$$c = \frac{1}{2d+1}$$

With this choice of c and since f is explosive, after a bit a fiddling, we find that

$f(m)$ eventually dominates $p(2m) + f(c(2dm))$.

This gives

$t > f(c(2d\|M^*\|) > f(c\|\Psi(M^*, M^*\|)$.

But t was the number of steps M took to decide that $\Psi(M^*, M^*)$ has $\mathcal{P}$.

Our Sketch Ends

It does no harm to suppose that $d > 1$. We take

$$c = \frac{1}{2d+1}$$

With this choice of c and since f is explosive, after a bit a fiddling, we find that

$f(m)$ eventually dominates $p(2m) + f(c(2dm))$.

This gives

$t > f(c(2d\|M^*\|) > f(c\|\Psi(M^*, M^*\|)$.

But t was the number of steps M took to decide that $\Psi(M^*, M^*)$ has $\mathcal{P}$. So our sketch is finished, if not complete.

Problems about equations

Let us fix a particular signature—a collection of operation symbols each with some fixed finite rank. We take $\mathbb{S}$ to be the set of all equations of our signature. Among the interesting choices for $\mathcal{P} \subseteq \mathbb{S}$ are the **equational theories**. These are just the sets of equations which are closed with respect to logical consequence.

Outline

Deciding equational theories

Except in the most meager signatures, it has been known at least since the work of Jan Kalicki in 1955 that there are uncountably many equational theories of a fixed signature. As there are only countably many computer programs, the greater part of these equational theories must be undecidable.

Deciding equational theories

But there are only countably many equational theories which a finitely axiomatizable (alias finitely based), so such cardinality arguments are silent about any finitely axiomatizable theory, as they are about any specific theory.

Finitely axiomatizable undecidable equational theories

1. There is a finitely axiomatizable undecidable equational theory of semigroups with several distinguished constants. [Post and Markov, 1947]
2. The equational theory of relation algebras is undecidable. [Tarski, 1948]
3. There is a finitely axiomatizable undecidable equational theory of groups with several additional distinguished constants. [Novikov and Boone, circa 1955]
4. There are finitely axiomatizable undecidable equational theories with just two one-place operation symbols or just one operation symbol (which is two-place). [Mal'cev, 1966]

Finitely axiomatizable undecidable equational theories

1. There is a finitely axiomatizable undecidable equational theory of semigroups with two distinguished constants. [Marshall Hall, 1949]
2. The equational theory of relation algebras is undecidable. [Tarski, 1948]
3. There is a finitely axiomatizable undecidable equational theory of groups with several additional distinguished constants. [Novikov and Boone, circa 1955]
4. There are finitely axiomatizable undecidable equational theories with just two one-place operation symbols or just one operation symbol (which is two-place). [Mal'cev, 1966]

Finitely axiomatizable undecidable equational theories

1. There is a finitely axiomatizable undecidable equational theory of semigroups with two distinguished constants. [Marshall Hall, 1949]
2. The equational theory of relation algebras is undecidable. [Tarski, 1948]
3. There is a finitely axiomatizable undecidable equational theory of groups with several additional distinguished constants. [Novikov and Boone, circa 1955]
4. There are finitely axiomatizable undecidable equational theories with just two one-place operation symbols or just one operation symbol (which is two-place). [Mal'cev, 1966]

Finitely axiomatizable undecidable equational theories

1. There is a finitely axiomatizable undecidable equational theory of semigroups with two distinguished constants. [Marshall Hall, 1949]
2. The equational theory of relation algebras is undecidable. [Tarski, 1948]
3. There is a finitely axiomatizable undecidable equational theory of groups with several additional distinguished constants. [Novikov and Boone, circa 1955]
4. There are finitely axiomatizable undecidable equational theories with just two one-place operation symbols or just one operation symbol (which is two-place). [Mal'cev, 1966]

Finitely axiomatizable undecidable equational theories

1. There is a finitely axiomatizable undecidable equational theory of semigroups with two distinguished constants. [Marshall Hall, 1949]
2. The equational theory of relation algebras is undecidable. [Tarski, 1948]
3. There is a finitely axiomatizable undecidable equational theory of groups with several additional distinguished constants. [Novikov and Boone, circa 1955]
4. There are finitely axiomatizable undecidable equational theories with just two one-place operation symbols or just one operation symbol (which is two-place). [Mal'cev, 1966]

More finitely axiomatizable undecidable equational theories

5. There is a finitely axiomatizable undecidable equational theory of semigroups. [Murskiĭ, 1968]
6. There is a finitely axiomatizable undecidable equational theory of Lie algebras with several additional distinguished constants. [Bo'kut, 1972]
7. There is a finitely axiomatizable undecidable equational theory of division rings (with the stipulation $0^{-1} \approx 0$) and with several additional distinguished constants. [Macintyre, 1973]
8. The equational theory of modular lattices is undecidable. [Freese, 1980]
9. There is a finitely axiomatizable undecidable equational theory of groups. [Yu. Kleĭman, 1982]

More finitely axiomatizable undecidable equational theories

5. There is a finitely axiomatizable undecidable equational theory of semigroups. [Murskiĭ, 1968]
6. There is a finitely axiomatizable undecidable equational theory of Lie algebras with several additional distinguished constants. [Bo'kut, 1972]
7. There is a finitely axiomatizable undecidable equational theory of division rings (with the stipulation $0^{-1} \approx 0$) and with several additional distinguished constants. [Macintyre, 1973]
8. The equational theory of modular lattices is undecidable. [Freese, 1980]
9. There is a finitely axiomatizable undecidable equational theory of groups. [Yu. Kleiman, 1982]

More finitely axiomatizable undecidable equational theories

5. There is a finitely axiomatizable undecidable equational theory of semigroups. [Murskiĭ, 1968]
6. There is a finitely axiomatizable undecidable equational theory of Lie algebras with several additional distinguished constants. [Bo'kut, 1972]
7. There is a finitely axiomatizable undecidable equational theory of division rings (with the stipulation $0^{-1} \approx 0$) and with several additional distinguished constants. [Macintyre, 1973]
8. The equational theory of modular lattices is undecidable. [Freese, 1980]
9. There is a finitely axiomatizable undecidable equational theory of groups. [Yu. Kleiman, 1982]

More finitely axiomatizable undecidable equational theories

5. There is a finitely axiomatizable undecidable equational theory of semigroups. [Murskiĭ, 1968]
6. There is a finitely axiomatizable undecidable equational theory of Lie algebras with several additional distinguished constants. [Bo'kut, 1972]
7. There is a finitely axiomatizable undecidable equational theory of division rings (with the stipulation $0^{-1} \approx 0$) and with several additional distinguished constants. [Macintyre, 1973]
8. The equational theory of modular lattices is undecidable. [Freese, 1980]
9. There is a finitely axiomatizable undecidable equational theory of groups. [Yu. Kleiman, 1982]

More finitely axiomatizable undecidable equational theories

5. There is a finitely axiomatizable undecidable equational theory of semigroups. [Murskiĭ, 1968]
6. There is a finitely axiomatizable undecidable equational theory of Lie algebras with several additional distinguished constants. [Bo'kut, 1972]
7. There is a finitely axiomatizable undecidable equational theory of division rings (with the stipulation $0^{-1} \approx 0$) and with several additional distinguished constants. [Macintyre, 1973]
8. The equational theory of modular lattices is undecidable. [Freese, 1980]
9. There is a finitely axiomatizable undecidable equational theory of groups. [Yu. Kleiman, 1982]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$, and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, co-NP complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, co-NP complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is co-NP complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

Decidable equational theories

1. The equational theory of any finite algebra is decidable and belongs to **co-NP**.
2. The equational theory of groups is decidable in polynomial time. [M. Dehn, 1912]
3. The equational theory of lattices is decidable in polynomial time. [T. Skolem, 1920]
4. The equational theory of each of the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R},$ and $\mathbb{C}$ is decidable.
5. The equational theory of Boolean algebras is decidable and, in fact, **co-NP** complete. [Cook, 1971]
6. The equational theory of any finite nontrivial lattice is decidable and, in fact, **co-NP** complete. [Bloniarz, Hunt, and Rosenkranz, 1987]
7. In particular, the equational theory of distributive lattices is **co-NP** complete. [Tschantz]

An observation of Ralph Freese

The equational theory of all lattices in decidable in polynomial time.

The equational theory of distributive lattices is decidable, but probably intractible—it is **co-NP** complete.

The equational theory of modular lattices is undecidable.

An observation of Ralph Freese

The equational theory of all lattices in decidable in polynomial time.

The equational theory of distributive lattices is decidable, but probably intractible—it is **co-NP** complete.

The equational theory of modular lattices is undecidable.

An observation of Ralph Freese

The equational theory of all lattices in decidable in polynomial time.

The equational theory of distributive lattices is decidable, but probably intractible—it is **co-NP** complete.

The equational theory of modular lattices is undecidable.

An observation of Ralph Freese

The equational theory of all lattices in decidable in polynomial time.

The equational theory of distributive lattices is decidable, but probably intractible—it is **co-NP** complete.

The equational theory of modular lattices is undecidable.

One last example

The equational theory of $\langle \mathbb{R}, +, \cdot, -, 1, \sin, |\ | \rangle$ is undecidable.

This was proven in 1993 by Yuri Matiyasevich using the negative solution to Hilbert's 10^{th} Problem and some ideas originating with Daniel Richardson in 1968.

Equational theories, overall

It seems that undecidable equational theories arise largely by design (usually with considerable effort). They seem pathological—most unlikely to be encountered in the course of mathematical practice.

Among the theories listed above, the equational theories of relation algebras and of modular lattices are striking exceptions to this pathology, as is the equational theory of $\langle \mathbb{R}, +, \cdot, -, 1, \sin, |\ | \rangle$.

Thus, the situation for equational theories with respect to decidability differs sharpley from the situation for elementary theories—and even universal Horn theories.

Equational theories, overall

It seems that undecidable equational theories arise largely by design (usually with considerable effort). They seem pathological—most unlikely to be encountered in the course of mathematical practice.

Among the theories listed above, the equational theories of relation algebras and of modular lattices are striking exceptions to this pathology, as is the equational theory of $\langle \mathbb{R}, +, \cdot, -, 1, \sin, |\ | \rangle$.

Thus, the situation for equational theories with respect to decidability differs sharpley from the situation for elementary theories—and even universal Horn theories.

Equational theories, overall

It seems that undecidable equational theories arise largely by design (usually with considerable effort). They seem pathological—most unlikely to be encountered in the course of mathematical practice.

Among the theories listed above, the equational theories of relation algebras and of modular lattices are striking exceptions to this pathology, as is the equational theory of $\langle \mathbb{R}, +, \cdot, -, 1, \sin, |\ | \rangle$.

Thus, the situation for equational theories with respect to decidability differs sharpley from the situation for elementary theories—and even universal Horn theories.

Outline

Properties of finite sets of equations

Now let us take $\mathbb{S}$ to be the collection of all finite sets of equations for some fixed signature.

We might want to know of a finite set Σ of equations

- whether Σ is true in some nontrivial finite algebra, or
- whether Σ is a set of axioms for group theory, or
- whether the variety Σ axiomatizes is congruence modular or
- ....

Properties of finite sets of equations

Now let us take $\mathbb{S}$ to be the collection of all finite sets of equations for some fixed signature.

We might want to know of a finite set Σ of equations

- whether Σ is true in some nontrivial finite algebra, or
- whether Σ is a set of axioms for group theory, or
- whether the variety Σ axiomatizes is congruence modular or
- ....

Properties of finite sets of equations

Now let us take $\mathbb{S}$ to be the collection of all finite sets of equations for some fixed signature.

We might want to know of a finite set Σ of equations

- whether Σ is true in some nontrivial finite algebra, or
- whether Σ is a set of axioms for group theory, or
- whether the variety Σ axiomatizes is congruence modular or
- . . . .

Properties of finite sets of equations

Now let us take $\mathbb{S}$ to be the collection of all finite sets of equations for some fixed signature.

We might want to know of a finite set Σ of equations

- whether Σ is true in some nontrivial finite algebra, or
- whether Σ is a set of axioms for group theory, or
- whether the variety Σ axiomatizes is congruence modular or
- ....

Properties of finite sets of equations

Now let us take $\mathcal{S}$ to be the collection of all finite sets of equations for some fixed signature.

We might want to know of a finite set Σ of equations

- whether Σ is true in some nontrivial finite algebra, or
- whether Σ is a set of axioms for group theory, or
- whether the variety Σ axiomatizes is congruence modular or
- ....

All of these problems turn out to be undecidable. In contrast to the situation for equational theories, the properties ordinarily encountered in mathematical practice and which are undecidable turn out to be plentiful.

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,. . . . [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,. . . . [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,. . . . [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,. . . . [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,.... [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,.... [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,.... [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Some undecidable properties of finite sets of equations

1. Whether Σ has a nontrivial model. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
2. Whether Σ has a nontrivial finite model. [McKenzie, 1975]
3. Whether Σ axiomatizes a minimal variety. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
4. Whether Σ axiomatizes a decidable equational theory. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
5. Whether Σ axiomatizes the theory of a finite algebra. [Perkins, 1966, McNulty, Murskiĭ circa 1970]
6. Whether Σ axiomatizes a variety with the amalgamation property. [Pigozzi, 1974]
7. Whether Σ axiomatizes a variety which is congruence modular,.... [McNulty,1972]
8. Whether Σ axiomatizes a variety which has near-unanimity terms. [McNulty, 1972]

Base undecidable equational theories

An equational theory T is called **base undecidable** provided there is no agorithm for determining of a finite set Σ of equations whether Σ axiomatizes T. For convenience, we say a term t is **nontrivial** when some operation symbol of rank at least 2 occurs in t or when at least two different operation symbols of rank 1 occur in t.

Base undecidable equational theories

The Base Undecidability Theorem

If T is a finitely axiomatizable equational theory and there is a nontrivial term t such that $t \approx x \in T$, then T is base undecidable.

This theorem was found around 1970 independently by McNulty and Murskiĭ.

Base undecidable equational theories

The Base Undecidability Theorem

If T is a finitely axiomatizable equational theory and there is a nontrivial term t such that $t \approx x \in T$, then T is base undecidable.

This theorem was found around 1970 independently by McNulty and Murskiĭ.

According to this theorem, the great bulk of finitely based equational theories arising in mathematical practice are base undecidable. But this does not apply to the equational theory of semigroups—to give one important exception.

Compatibility with topological spaces

Let $\mathbb{T}$ be a topological space—for example, the real line with its usual topology. An algebra $\mathbf{T}$ with universe $\mathbb{T}$ is called a **topological algebra** if all the basic operations of $\mathbf{T}$ are continuous. A set Σ of equations is **compatible** with $\mathbb{T}$ provided there is a topological algebra $\mathbf{T}$ so that Σ is true in $\mathbf{T}$.

Compatibility with topological spaces

Taylor's Compatibility Theorem

Fix a signature adequate for the theory of rings with unit expanded by two additional 1-place operation symbols. There is no algorithm for deciding whether an equation of this signature is compatible with the real line.

Walter Taylor proved this in 2006 in a slightly weaker form.

Compatibility with topological spaces

On the other hand, for any finite signature, Walter Taylor observed that for each of the spheres $\mathbb{S}^n$ there is an algorithm for determining of finite sets Σ of equations, whether Σ is compatible with $\mathbf{S}^n$, as long as $n \neq 1, 3, 7$.

Compatibility with topological spaces

On the other hand, for any finite signature, Walter Taylor observed that for each of the spheres $\mathbb{S}^n$ there is an algorithm for determining of finite sets Σ of equations, whether Σ is compatible with $\mathbf{S}^n$, as long as $n \neq 1, 3, 7$. The complexities of these compatibility problems for $\mathbb{S}^n$ have not been worked out.

Compatibility with topological spaces

On the other hand, for any finite signature, Walter Taylor observed that for each of the spheres $\mathbb{S}^n$ there is an algorithm for determining of finite sets Σ of equations, whether Σ is compatible with $\mathbf{S}^n$, as long as $n \neq 1, 3, 7$. The complexities of these compatibility problems for $\mathbb{S}^n$ have not been worked out. Also, there is some reason to think that the compatibility problem for $\mathbb{S}^1$ might also turn out to be decidable. The situation for $\mathbb{S}^3$ and $\mathbb{S}^7$ and for other common topological spaces seems to be wide open.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

In the first two cases, the compatibility problem is known to be **co-NP** complete.

Compatibility with equational theories

Let T be some equational theory. We say that a set Σ of equations is **compatible** with T when $\Sigma \cup T$ has a nontivial model. For example, take T to be the equational theory of lattices. Then Σ is compatible with T when Σ holds in some nontrivial lattice. We say that T is **compatibility decidable** provided there is an algorithm which upon input of any finite set Σ of equations will determine whether Σ is compatible with T.

It was observed in the 1960's that each of the following equational theories is compatibility decidable.

1. The equational theory of Boolean algebras.
2. The equational theory of lattices.
3. The equational theory of groups.
4. The equational theory of rings.

In the first two cases, the compatibility problem is known to be **co-NP** complete.The last two cases deserve some attention.

Our Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Our Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

Our Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

How hard must that be?

Our Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

How hard must that be?

This problem is **co-NP** complete.

Our Sample Problem

Is there a method which will determine of any given equation whether it is true in some nontrivial lattice?

Just check whether the equation is true in the 2-element lattice.

How hard must that be?
This problem is **co-NP** complete.
Yes, but can it be done in polynomial time? Send the answer to the Clay Foundation.

The set up

Now we consider problems where the input is a finite algebra. To ensure that an appropriate complexity function in available, we will insist that each finite algebra have only finitely many fundamental operations. This will allow us to enter each finite algebra via a computer keyboard.

The set up

Now we consider problems where the input is a finite algebra. To ensure that an appropriate complexity function in available, we will insist that each finite algebra have only finitely many fundamental operations. This will allow us to enter each finite algebra via a computer keyboard.

Infinite algebras or even finite algebras with infinitely many fundamental operations are not directly amendable as inputs. While we will not pursue this avenue, the assortment of such infinite algebras that still admit finite descriptions of one kind or another is rich and mathematically significant. Instead of framing computational problems about these algebras, one instead frames them about the finite descriptions. That is, the inputs are the descriptions rather than the algebras.

Outline

The Finite Algebra Membership Problem

Let $\mathcal{V}$ be a variety (other kinds of classes are interesting too) of some fixed finite signature ρ. Then the Finite Algebra Membership Problem for $\mathcal{V}$ is

INSTANCE: A finite algebra $\mathbf{B}$ *of signature* ρ*.*

QUESTION: Is $\mathbf{B} \in \mathcal{V}$*?*

The Finite Algebra Membership Problem

Let $\mathcal{V}$ be a variety (other kinds of classes are interesting too) of some fixed finite signature ρ. Then the Finite Algebra Membership Problem for $\mathcal{V}$ is

INSTANCE: A finite algebra $\mathbf{B}$ *of signature* ρ*.*

QUESTION: Is $\mathbf{B} \in \mathcal{V}$*?*

In the event that $\mathcal{V}$ is finitely axiomatizable, it is not hard to see that this problem can be solved in polynomial time.

The Finite Algebra Membership Problem

Let us consider the case when $\mathcal{V}$ is generated by a finite algebra $\mathbf{A}$.

The first advance was the observation made by Jan Kalicki in 1952 that there is an algorithm which, upon input of finite algebras $\mathbf{A}$ and $\mathbf{B}$, will determine whether $\mathbf{B}$ belongs to the variety generated by $\mathbf{A}$. Kalicki's algorithm is a brute force affair.

The Finite Algebra Membership Problem

Let us consider the case when $\mathcal{V}$ is generated by a finite algebra $\mathbf{A}$.

The first advance was the observation made by Jan Kalicki in 1952 that there is an algorithm which, upon input of finite algebras $\mathbf{A}$ and $\mathbf{B}$, will determine whether $\mathbf{B}$ belongs to the variety generated by $\mathbf{A}$. Kalicki's algorithm is a brute force affair.

Bergman and Słutzki, in 2000, gave a detailed analysis of a polished version of Kalicki's brute force algorithm, finding that it can be carried out in **2EXPTIME**.

The Finite Algebra Membership Problem

Consider the Finite Algebra Membership Problem where $\mathcal{V}$ is generated by a finite algebra. Here are the recent findings:

1. A finite algebra with an **NP**-complete Finite Algebra Membership Problem. [Z. Székely, 1998]
2. A finite semigroup with an **NP**-hard Finite Algebra Membership Problem. [Jackson and McKenzie, 2006]
3. A finite algbra with a **2EXPTIME**-complete Finite Algebra Membership Problem. [Kozik, 2007]

The Finite Algebra Membership Problem

Consider the Finite Algebra Membership Problem where $\mathcal{V}$ is generated by a finite algebra. Here are the recent findings:

1. A finite algebra with an **NP**-complete Finite Algebra Membership Problem. [Z. Székely, 1998]
2. A finite semigroup with an **NP**-hard Finite Algebra Membership Problem. [Jackson and McKenzie, 2006]
3. A finite algbra with a **2EXPTIME**-complete Finite Algebra Membership Problem. [Kozik, 2007]

The Finite Algebra Membership Problem

Consider the Finite Algebra Membership Problem where $\mathcal{V}$ is generated by a finite algebra. Here are the recent findings:

1. A finite algebra with an **NP**-complete Finite Algebra Membership Problem. [Z. Székely, 1998]
2. A finite semigroup with an **NP**-hard Finite Algebra Membership Problem. [Jackson and McKenzie, 2006]
3. A finite algbra with a **2EXPTIME**-complete Finite Algebra Membership Problem. [Kozik, 2007]

Outline

Tarski's Finite Basis Problem

An algebra or a variety is **finitely based** if its equational theory is finitely axiomatizable. Tarski's Finite Basis Problem is

INSTANCE: A finite algebra $\mathbf{A}$.

QUESTION: Is $\mathbf{A}$ *finitely based?*

Tarski's Finite Basis Problem

An algebra or a variety is **finitely based** if its equational theory is finitely axiomatizable. Tarski's Finite Basis Problem is

INSTANCE: A finite algebra $\mathbf{A}$.

QUESTION: Is $\mathbf{A}$ *finitely based?*

Tarski raised the question of whether there is an alogrithm to solve this problem in the early 1960's.

Tarski's Finite Basis Problem

Finite algebras belonging to many familiar varieties are now known to be finitely based. This applies to groups, rings, lattices, Boolean algebras, Lie algebras, and many others. So restricted to classes like these, there is an easy algorithm for solving Tarski's Finite Basis Problem.

Tarski's Finite Basis Problem

Observe that Tarski's Finite Basis Problem places no restriction on the signature of the input algebras. In 1984, Ralph McKenzie proved that restricting the signature to just one operation symbol, that one being 2-place, results a problem equivalent to the original in the sense that there is an algorithm solving one if and only if there is an algorithm solving the other.

Tarski's Finite Basis Problem

A finite algebra is **inherently nonfinitely based** if it belongs to no locally finite finitely based variety.

In 1987 Mark Sapir discovered an algorithm to solve the following

INSTANCE: A finite semigroup **A**.

QUESTION: Is **A** *inherently nonfinitely based?*

Tarski's Finite Basis Problem

Finally, in 1996 Ralph McKenzie published a resolution of Tarski's Finite Basis Problem: It is undecidable.

Outline

Residual bounds

A variety $\mathcal{V}$ has a **finite residual bound** provided there is a natural number k such that every subdirectly irreducible algebra in $\mathcal{V}$ has cardinality less than k.

Residual bounds

A variety $\mathcal{V}$ has a **finite residual bound** provided there is a natural number k such that every subdirectly irreducible algebra in $\mathcal{V}$ has cardinality less than k.

McKenzie also showed that the following problem is undecidable:

INSTANCE: A finite algebra $\mathbf{A}$.

QUESTION: Does the variety generated by $\mathbf{A}$ *have a finite residual bound?*

Minimal varieties

A variety $\mathcal{V}$ is **minimal** if $\mathcal{V}$ has exactly one proper subvariety.

INSTANCE: A finite algebra $\mathbf{A}$.

QUESTION: Is the variety generated by $\mathbf{A}$ *minimal?*

Minimal varieties

A variety $\mathcal{V}$ is **minimal** if $\mathcal{V}$ has exactly one proper subvariety.

INSTANCE: A finite algebra **A**.

QUESTION: Is the variety generated by **A** *minimal?*

In 1956, Dana Scott gave an algorithm for deciding this question. Scott's algorithm is a brute force algorithm which invokes Kalicki algorithm mentioned above. The computational complexity of this problem is open.

Congruence distributivity and its relatives

Properties of a variety like congruence permutability, congruence distributivity, congruence modularity, . . . , have been shown to be Mal'cev properties. In many cases, the proof that they are Mal'cev properties provides an additional equivalent condition.

Congruence distributivity and its relatives

Properties of a variety like congruence permutability, congruence distributivity, congruence modularity, . . . , have been shown to be Mal'cev properties. In many cases, the proof that they are Mal'cev properties provides an additional equivalent condition.

A variety $\mathcal{V}$ is congruence distributive if and only if the algebra in $\mathcal{V}$ which is $\mathcal{V}$-freely generated by 3 elements has distributive congruences.

Congruence distributivity and its relatives

This means that the problem below is decidable (in those cases)

INSTANCE: A finite algebra $\mathbf{A}$*.*

QUESTION: Is the variety generated by $\mathbf{A}$ *congruence distributive (or permutable, modular,. . .)?*

Congruence distributivity and its relatives

This means that the problem below is decidable (in those cases)

INSTANCE: A finite algebra $\mathbf{A}$*.*

QUESTION: Is the variety generated by $\mathbf{A}$ *congruence distributive (or permutable, modular,. . .)?*

The computational complexity of this kind of problem will be addressed in the talk of Ralph Freese slated for Wednesday afternoon at 2:00 p.m.

Near unanimity terms

A variety $\mathcal{V}$ has a **near unanimity term** provided there is a term $t(x_0, x_1, \ldots, x_{n-1})$ in which at least three distinct variable occur such that

$$t(y, x, x, \ldots, x) \approx t(x, y, x, \ldots, x) \approx$$
$$t(x, x, y, \ldots, x) \approx \cdots \approx t(x, x, x, \ldots, y) \approx x$$

holds in $\mathcal{V}$. Lattice, for example, have a 3-place near unanimity term.

Miklós Maróti will speak to us Friday morning about the following problem:

INSTANCE: A finite algebra **A**.

QUESTION: Does the variety generated by **A** *have a near unanimity term?*

Affine completeness

An algebra $\mathbf{A}$ is called **affine complete** if each finitary operation on A which is compatible with all the congruences of $\mathbf{A}$ is a polynomial of $\mathbf{A}$. A variety is affine complete if each algebra belonging to the variety is affine complete.

In 2002, Kaarli and Pixley gave an algorithm to solve the following problem.

> *INSTANCE: A finite algebra* $\mathbf{A}$*.*
>
> *QUESTION: Is the variety generated by* $\mathbf{A}$ *affine complete?*

Affine completeness

An algebra **A** is called **affine complete** if each finitary operation on A which is compatible with all the congruences of **A** is a polynomial of **A**. A variety is affine complete if each algebra belonging to the variety is affine complete.

In 2002, Kaarli and Pixley gave an algorithm to solve the following problem.

> *INSTANCE: A finite algebra* **A**.
>
> *QUESTION: Is the variety generated by* **A** *affine complete?*

The computational complexity of this problem is unknown.

Primality and quasiprimality

Let $\mathcal{A}$ be an algebra. The clone of term-functions of $\mathbf{A}$ is denoted by $\operatorname{Clo}\mathbf{A}$. The algebra $\mathbf{A}$ is **primal** provided $\operatorname{Clo}\mathbf{A}$ is the set of all finitary operations on the set A. Primal algebras must be finite. The 2-element Boolean algebra is primal.

The **ternary discriminator operation** on the set A is the function $d : A^3 \to A$ so that

$$d(a,b,c) = \begin{cases} a & \text{if } a \neq b \\ c & \text{if } a = b. \end{cases}$$

$\mathbf{A}$ is said to be **quasiprimal** provided A is finite and the ternary discriminator belongs to $\operatorname{Clo}\mathbf{A}$.

Primality and quasiprimality

The problems

INSTANCE: A finite algebra $\mathbf{A}$.

QUESTION: Is $\mathbf{A}$ *primal (or quasiprimal)?*

It follows from the 1964 work of Alfred Foster and Alden Pixley, and of Pixley in 1971 that both of these problems have algoritmic solutions. However, little seems to be known about their computational complexity.

Dualizability

Some finite algebras **A**, like the 2-element Boolean algebra, provide a natural duality between the quasivariety generated by **A** and a corresponding class of (structured) topological spaces. Time does not permit me to give here a full definition of what it means for **A** to be **dualizable**. However, anyone who googles the name of Brian Davey will find out rapidly what is what.

Consider the problem

INSTANCE: A finite algebra **A**.

QUESTION: Is **A** *dualizable?*

Dualizability

Like the finite basis property and the existence of a finite residual bound, dualizability appears to be a strong finiteness property of (in this case) the quasivariety generated by the algebra. Roughly speaking, for most of the properties of a finite algebra $\mathbf{A}$ which turn out to be decidable, the algorithms seem to depend on examining the direct powers $\mathbf{A}^n$ up to some finite n (which might depend on $\mathbf{A}$). Dualizability appears to require the examination of all the finite direct powers of $\mathbf{A}$. While there is some prospect that this problem will turn out to be undecidable,

Dualizability

Like the finite basis property and the existence of a finite residual bound, dualizability appears to be a strong finiteness property of (in this case) the quasivariety generated by the algebra. Roughly speaking, for most of the properties of a finite algebra $\mathbf{A}$ which turn out to be decidable, the algorithms seem to depend on examining the direct powers $\mathbf{A}^n$ up to some finite n (which might depend on $\mathbf{A}$). Dualizability appears to require the examination of all the finite direct powers of $\mathbf{A}$. While there is some prospect that this problem will turn out to be undecidable, no one knows!

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?

Some open problems

Problem 1. Is it decidable of a finite set of equations whether it is compatible with the topological space $\mathbb{S}^1$?

Problem 2. What is the complexity of deciding of a finite set of equations whether it is compatible with $\mathbb{S}^n$ for $n \neq 1, 3, 7$?

Problem 3. What is the complexity of deciding of a finite set of equations whether it is true in some nontrivial group (or ring)?

Problem 4. What is the complexity of deciding whether a finite algebra generates a minimal variety?

Problem 5. What is the complexity of deciding whether a finite algebra has a near unanimity term?

Problem 6. What is the complexity of deciding whether a finite algebra generates an affine complete variety?

Problem 7. What is the complexity of deciding whether a finite algebra is primal? Quasiprimal?

Problem 8. Is it decidable whether a finite algebra is dualizable?