Why Think?

Letting Computers Do Math For Us

Chris Grossack
(they/them)

January 22, 2021

Syntax

- In logic we always work relative to some allowable symbols
- There are Primitive Symbols related to our objects of interest
- There are Logical Symbols related to the type of logic we're studying
- Groups have primitive symbols $\cdot$, $^{-1}$, and 1
- Partial Orders have primitive symbols $\leq$
- First Order Logic (FO) allows $\wedge$, $\neg$, $\forall x$, $\exists x$, etc.
- Infinitary Logic ($\mathrm{FO}_{\omega_1,\omega}$) allows $\wedge$, $\neg$, $\bigwedge$, $\forall x$, $\exists x$, etc.
- Monadic Second Order Logic (MSO) allows $\wedge$, $\neg$, $\forall x$, $\exists x$, $\forall X$, $\exists X$, etc.

Here x means an element of the domain, and X means a subset of the domain

Formulas

If we have a logic $\mathcal{L}$ (for instance, first order logic) and some primitive symbols σ, we write $\mathcal{L}(\sigma)$ for the set of all formulas you can write using the logical symbols and the primitive symbols.
A formula is called a sentence if it has no free variables.

- $\forall x . x \cdot 1 = x \wedge 1 \cdot x = x$ is a sentence in $\mathrm{FO}(\cdot, {}^{-1}, 1)$
- $\forall x . \bigvee_{n \in \mathbb{N}} x = S^n 0$ is a sentence in $\mathrm{FO}_{\omega_1, \omega}(S, 0)$
- $\forall A . \exists x \in A . \forall y \in A . x \leq y$ is a sentence in $\mathrm{MSO}(\leq)$
- $xy = yx$ is a formula in $\mathrm{FO}(\cdot, {}^{-1}, 1)$ with 2 free variables.

Theories

A subset of $\mathcal{L}(\sigma)$ is called a Theory.

For instance, the theory of groups is given by

$$\left\{ \text{All sentences provable from } \begin{array}{l} \forall x.\forall y.\forall z.(xy)z = x(yz) \\ \forall x.x1 = x \wedge 1x = x \\ \forall x.x^{-1}x = 1 \wedge xx^{-1} = 1 \end{array} \right\}$$

Semantics

Given a set X (with functions/relations interpreting the symbols in σ) and a $\mathcal{L}(\sigma)$ theory T, we say that $X \models T$ whenever X thinks that every sentence in T is true.

Decidable Theories

A theory T is called Decidable iff there is a computer program which takes a sentence φ as input and outputs YES (resp. NO) exactly when $\varphi \in T$ (resp. $\varphi \notin T$)

Such a program is called a Decider for T.

Obvious Questions

- Which Theories are Decidable?
- How long does it take a decider to run?

We'll focus on the first question in this talk.

The short answer to the second question is "a long time, usually".

There's a useful lemma which will provide some inspiration for decidability results:

Lemma

Given any "computably enumerable" set of axioms, the set of theorems provable from those axioms is *also* computably enumerable

Proof.

List the conclusion of all proofs of length 1 using axiom 1. Then all proofs of length 2 with axioms 1 and 2. Then all proofs of length 3 with axioms 1, 2, and 3. And so on.
Every proof appears in this list, and every theorem is the conclusion of one of these proofs. □

- Here a set is computably enumerable whenever a computer program can *list* all of the elements of the set.
- Notice since we don't know where an element will turn up, we can't tell if an element *isn't* in the list.

Definition

A thoery T is called Complete if exactly one of φ or $\neg\varphi$ is in T. That is, T should have an opinion on every sentence φ.

Theorem

If T is a complete theory with a computably enumerable set of axioms, then T is decidable.

Proof.

If T has a computably enumerable set of axioms, then we can list all its theorems. But since it's complete, either φ or $\neg\varphi$ will show up on the list. Wait until you see one of them, and then you'll know if T thinks φ is true or false. $\square$

Ok. So complete axiomatizable theories are decidable. What does that mean, practically?

If you have any question about

- ACF$_p$: Algebraically closed fields of fixed characteristic (with $+, \times, 0, 1$)
- DLO: Dense Linear Orders without Endpoints (think $(\mathbb{Q}, <)$)
- $(\mathbb{R}, +, \times, 0, 1, \leq)$
- Any fixed finite group

a computer can tell you the answer.

Let's sketch some techniques for seeing this. We'll move in increasing order of simplicity.

ACF$_p$ is complete

The idea is to use some heavy machinery from model theory.

The "Łoś-Vaught Test" says that if for some cardinal κ our theory T has exactly one model of cardinality κ (up to isomorphism), then T must be complete.

Since algebraically closed fields are determined up to isomorphism by their characterstic and transcendence degree, any two uncountable models of ACF$_p$ with cardinality κ are isomorphic (as their transcendence degree equals κ too).

Then, by the Łoś-Vaught test, ACF$_p$ is complete.

DLO is complete

We could argue by the Łoś-Tarski test again, as any countable model of DLO is isomorphic to $(\mathbb{Q}, <)$. But in the interest of showing more ideas, we'll show another strategy: Quantifier Elimination.

If we can give an algorithm that converts any sentence into an equivalent sentence without quantifiers, then we'll be done. Why? Notice the only quantifier free sentences in DLO are $\top$ and $\bot$!

Since each φ is equivalent to $\top$ or $\bot$ (and DLO has an opinion about those sentences) we see that DLO has an opinion about *every* sentence, and so it is complete.

Ok. So how do we eliminate quantifiers?

Quantifier Elimination for DLO

Now combinatorics saves the day! Every sentence can be rewritten as a collection of $<$s and $\leq$s joined by $\wedge$s and $\vee$s. Liberally applying DeMorgan's laws, and using $\exists x.(\varphi \vee \psi)$ is equivalent to $(\exists x \varphi) \vee (\exists x.\psi)$, it suffices to eliminate quantifiers from a sentence of the form

$$\exists x. \bigwedge_i y_i \leq x \wedge \bigwedge_j y_j < x \wedge \bigwedge_k x \leq z_k \wedge \bigwedge_l x < z_l$$

But a moment's thought (or perhaps many moments of thought...) shows this is equivalent to

$$\bigwedge_{i,k} y_i \leq z_k \wedge \bigwedge_{i,l} y_i < z_l \wedge \bigwedge_{j,k} y_j < z_k \wedge \bigwedge_{j,l} y_j < z_l$$

which is quantifier free.

Theory of a finite group is complete

Last and easiest is the theory of some fixed finite group. Let's say $\mathbb{Z}/3$. That is, look at $\{\varphi \mid \mathbb{Z}/3 \models \varphi\}$

Then this theory is complete, because $\mathbb{Z}/3$ has an opinion on every sentence. Moreover, it's finitely axiomatizable. After all, we can just write down the entire multiplication table.

It's not too surprising that a finite structure should have decidable theory. After all, there's only finitely many things to check for each quantifier, so we can just try them all.

Perhaps *more* surprising is that we get decidability for infinite structures as well, provided their functions and relations are "simple".

Here "simple" means that there is a rule that lets you compute all of the operations. Think of the addition algorithm for $\mathbb{Z}$. We formalize this with the notion of an Automatic Structure.

Definition: Automaticity

A structure X with operations $\{f_i\}$ and relations $\{R_j\}$ is called Automatic if it is isomorphic to a set of words over some alphabet, where each f_i and R_j can be computed by a finite state automaton.

- The actual definition of an automaton is rather unenlightening
- I'll give a definition by example on the next slide. It should be enough to understand roughly what's going on.
- The example to keep in mind is $(\mathbb{Z}, 0, +, \leq)$ – so let's see it.

- We first have to identify $\mathbb{Z}$ with the set of words over some alphabet.
- We could use $\{-,0,1,2,3,4,5,6,7,8,9\}$, but we'll use $\{-,0,1\}$ instead.
- We will represent our numbers by their binary expansion, possibly with a negative sign at the front.
- There is one minor technicality, though:
- We usually ask for our automata to read the strings from left to right.
- For some asinine reason, though, we add our numbers from right to left.
- Rather than modify the definition of automata that I never gave, we will retroactively fix this historical mistake, and we will write the least significant place on the left.
- So 8 is 0001, -6 is -011, and so on.

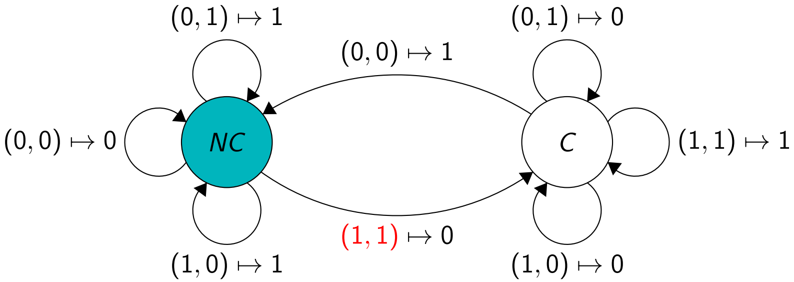
In the interest of simplicity, this machine will only work for positive numbers.

Moreover, it is a slightly simpler kind of machine than is often used for these theorems (for experts: we present a Mealy Automaton rather than a Büchi Automaton).

However it still showcases the idea of an automatic structure.

$$\begin{tikzcd}
NC \arrow[loop left, "(0,0)\mapsto 0"] \arrow[loop above, "(0,1)\mapsto 1"] \arrow[loop below, "(1,0)\mapsto 1"] \arrow[r, bend right, "(1,1)\mapsto 0"'] & C \arrow[l, bend right, "(0,0)\mapsto 1"'] \arrow[loop above, "(0,1)\mapsto 0"] \arrow[loop right, "(1,1)\mapsto 1"] \arrow[loop below, "(1,0)\mapsto 0"]
\end{tikzcd}$$

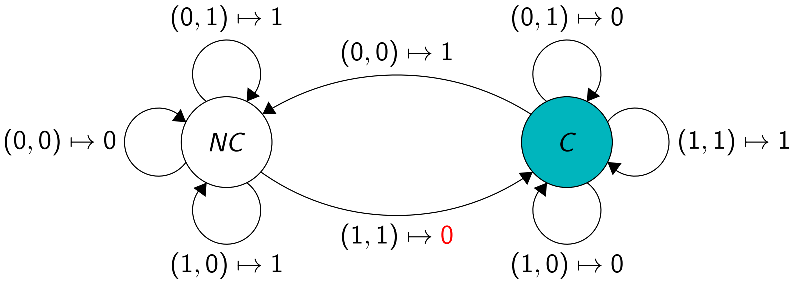
Let's use this to compute, say $1101 + 1001$



$(0,1) \mapsto 1$
$(0,0) \mapsto 1$
$(0,1) \mapsto 0$
$(0,0) \mapsto 0$
NC
C
$(1,1) \mapsto 1$
$(1,1) \mapsto 0$
$(1,0) \mapsto 1$
$(1,0) \mapsto 0$

1101

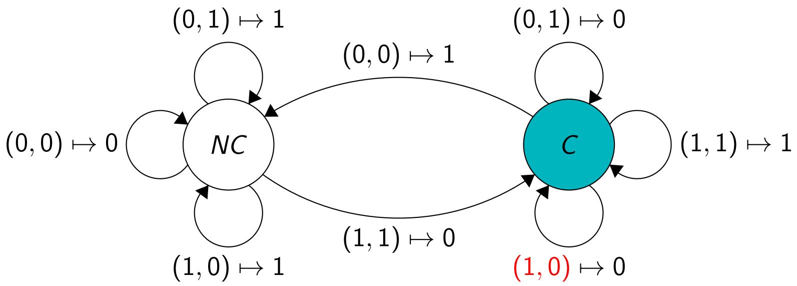
1001


$(0,1) \mapsto 1$
$(0,0) \mapsto 1$
$(0,1) \mapsto 0$
$(0,0) \mapsto 0$
NC
C
$(1,1) \mapsto 1$
$(1,1) \mapsto 0$
$(1,0) \mapsto 1$
$(1,0) \mapsto 0$

1101

1001

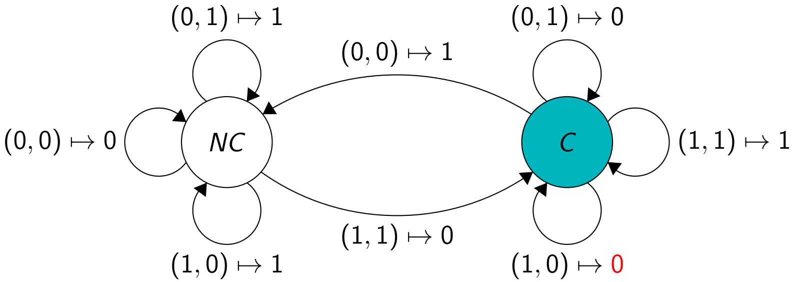
0


$(0,1) \mapsto 1$
$(0,1) \mapsto 0$
$(0,0) \mapsto 1$
$(0,0) \mapsto 0$
NC
C
$(1,1) \mapsto 1$
$(1,1) \mapsto 0$
$(1,0) \mapsto 1$
$(1,0) \mapsto 0$

1101

1001

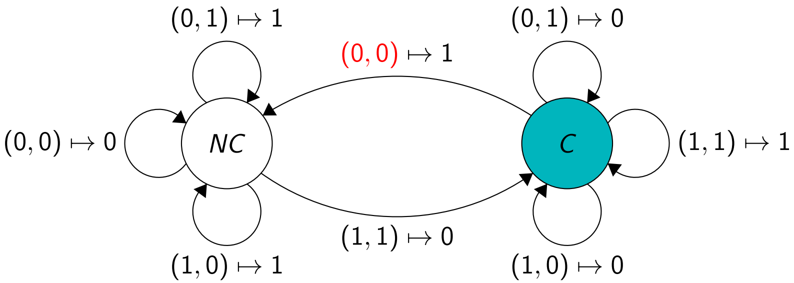
0


$(0,1) \mapsto 1$
$(0,1) \mapsto 0$
$(0,0) \mapsto 1$
$(0,0) \mapsto 0$
NC
C
$(1,1) \mapsto 1$
$(1,1) \mapsto 0$
$(1,0) \mapsto 1$
$(1,0) \mapsto 0$

1101

1001

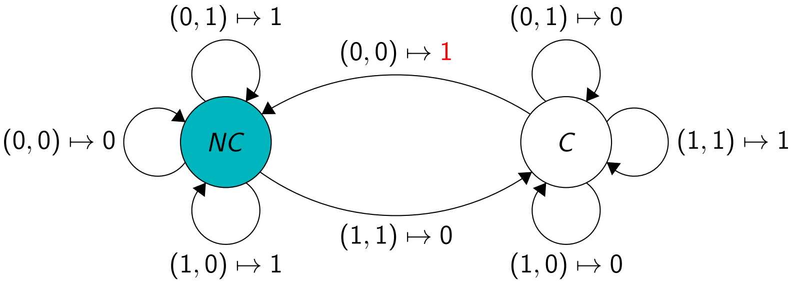
00



1101

1001

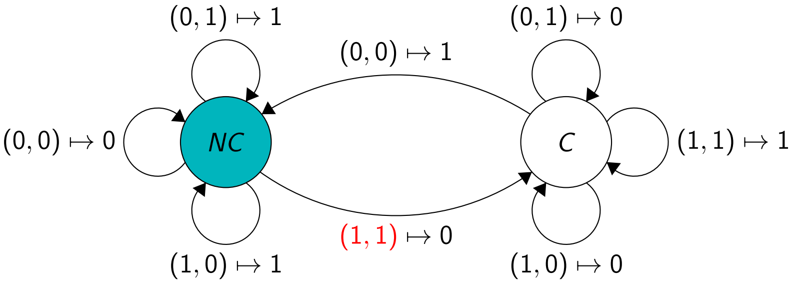
00



1101

1001

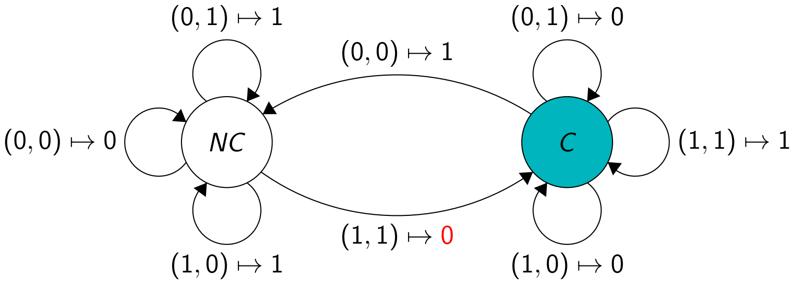
001



1101

1001

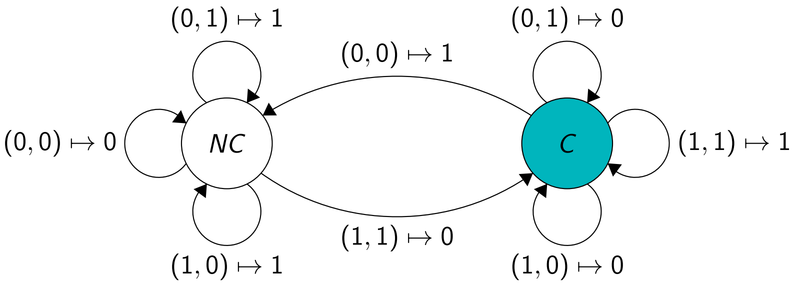
001



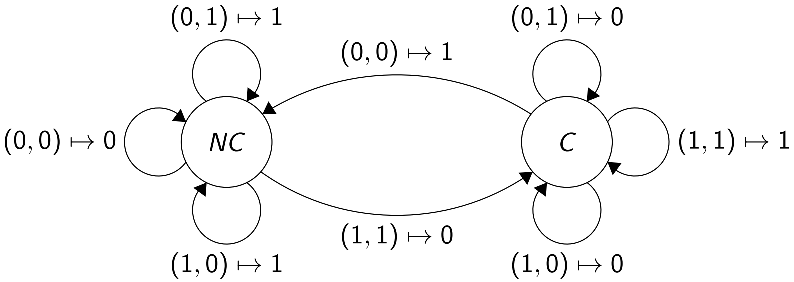
1101

1001

0010



1101
1001
00101



1101

1001

00101

Remembering our left-to-right rule, we see that we've successfully computed $11 + 9 = 20$.

Why do we care?

Theorem

If $\mathcal{A}$ is an automatic structure, then the theory of $\mathcal{A}$ is decidable.

In fact, even if we add quantifiers $\exists^{\infty} x$ and $\exists^{n \bmod m} x$ which say "there exist infinitely many x" and "there exist n mod m many x" respectively, the theory remains decidable.

So what are some automatic theories?

- $(\mathbb{Z}, +, 0, \leq)$ (Presburger Arithmetic)
- $(\mathbb{Z}, \times, 1, \leq)$ (Skolem Arithmetic)
- The MSO theory of $(\mathbb{N}, \leq)$.
- ⚠In geometric group theory, there is a notion of "automatic group" as defined by Thurston. This definition does *not* agree with Thurston's, and indeed there are counterexamples in both directions.

- So this is great and all
- But are we in risk of being put out of a job?
- Are the computer overlords coming?

Theorem

No.

Let's see some theories which are *not* decidable.

That is, theories for which no algorithm can possibly answer "is $\varphi \in T$".

- ZF(C) – any set theory worth its salt is undecidable
- Th$(\mathbb{N}, +, \times, 0, 1)$ – number theory is beyond computers too
- The theory of groups – rather interestingly, the theory of *abelian* groups is decidable!

All of these results basically rely on the Halting Problem which says there's no way to tell if a program will halt on a given input.

Here's an example

Theorem

Th($\mathbb{N}$) is not decidable

Proof.

sketch

1. Let M be a turing machine
2. Be clever
3. There is a formula $\varphi_M(x)$ so that $\varphi_M(n)$ is true if and only if M halts in $\leq n$ steps
    - Crucially, we need both $+$ and $\times$ to define $\varphi_M(x)$!
4. There is no algorithm to decide if $\exists n.\varphi_M(n)$, as otherwise you could solve the Halting Problem
5. So there is certainly no algorithm to decide the whole theory of $\mathbb{N}$.

□

Now that we have this, though, we can show other theories are undecidable by showing they can *simulate* $\mathbb{N}$.

$\mathrm{Th}(\mathbb{Z}, +, \times)$ isn't decidable

A theorem of Lagrange shows that every nonnegative integer can be written as a sum of 4 squares.

Then $\mathbb{N} = \{x \mid \exists a, b, c, d . a^2 + b^2 + c^2 + d^2 = x\} \subseteq \mathbb{Z}$

Then if we could answer every question about $\mathbb{Z}$, we could answer every question about $\mathbb{N}$ by asking it for $\mathbb{Z}$ and requiring that we only look at sums of four squares (which is expressible).

So, since $\mathbb{N}$ is undecidable, $\mathrm{Th}(\mathbb{Z})$ must be too.

As an interesting aside:

Not only is Th($\mathbb{Z}$) undecidable, the simpler question "does the polynomial $p \in \mathbb{Z}[x_1, \ldots, x_n]$ have an integer root?" is already undecidable!

Indeed, Hilbert's 10th problem asked if some algorithm could tell if a diophantine equation was solvable. Now we know the answer is no!

This is a famous theorem of Matiyasevich, building on the work of Putnam, Davis, and Robinson.

In fact, this was shown by showing something very cool! If you'll indulge me, we're close enough to it that I have to mention it.

MRDP Theorem

Every computably enumerable set of natural numbers is the solution set of a diophantine equation with parameters

Corollary

Since the halting problem (sufficiently coded) is computably enumerable, there is no algorithm to tell if a diophantine equation with parameters is solvable.

So Hilbert's 10th problem is unsolvable.

But what does this *mean*? Let's see some examples.

Let's start with the Pell Equation.

$$x^2 - N(y+1)^2 = 1$$

If you've ever taken a number theory class, you've probably seen that the equation $x^2 - Ny^2 = 1$ has solutions other than $(1,0)$ if and only if N is not a perfect square. By adding 1 to y, we remove this silly solution, and so we see the diophantine equation with parameters

$$\exists x, y. x^2 - N(y+1)^2 = 1$$

defines the computably enumerable set $\{N \mid N \text{ is not a perfect square}\}$

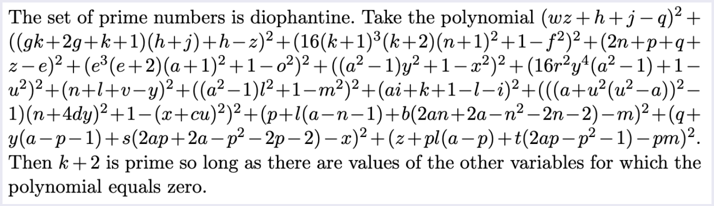
Similarly, $\exists x, y. a = (x+2)(y+2)$ determines $\{a \mid a \text{ is not prime}\}$

Matiyasevich promises we can find such an equation for *any* computably enumerable set. In fact, 9 variables suffices.

Has anyone found such a polynomial for the primes?

Yes! Buckle up, though...

Theorem


The set of prime numbers is diophantine. Take the polynomial $(wz+h+j-q)^2+((gk+2g+k+1)(h+j)+h-z)^2+(16(k+1)^3(k+2)(n+1)^2+1-f^2)^2+(2n+p+q+z-e)^2+(e^3(e+2)(a+1)^2+1-o^2)^2+((a^2-1)y^2+1-x^2)^2+(16r^2y^4(a^2-1)+1-u^2)^2+(n+l+v-y)^2+((a^2-1)l^2+1-m^2)^2+(ai+k+1-l-i)^2+(((a+u^2(u^2-a))^2-1)(n+4dy)^2+1-(x+cu)^2)^2+(p+l(a-n-1)+b(2an+2a-n^2-2n-2)-m)^2+(q+y(a-p-1)+s(2ap+2a-p^2-2p-2)-x)^2+(z+pl(a-p)+t(2ap-p^2-1)-pm)^2$. Then $k+2$ is prime so long as there are values of the other variables for which the polynomial equals zero.

You'll forgive me for not retyping that.

According to wikipedia, you can bring the number of variables down to 9 provided you're happy with a polynomial of degree $\sim 10^{45}$.

Conversely, you can get the degree down to 4 if you're willing to use 58 variables.

So this poses a rather interesting question:

- We know the theory of $(\mathbb{R}, +, \times, 0, 1)$ is decidable.
- So there's an algorithm which decides if a polynomial equation (with parameters) is solvable over $\mathbb{R}$.
- But we've just seen that there is no such algorithm to solve polynomials over $\mathbb{Z}$...
- What about $\mathbb{Q}$?

It's time for the $\sim \star \sim$ Interactive Portion $\sim \star \sim$ of the talk.

Say in chat if you think an algorithm can find rational roots!

This is *Super* open!

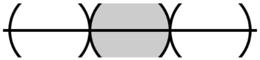
Very little is known. It's a really exciting avenue of research right now, with useful tools from number theory, algebraic geometry, model theory, computability theory, and lots more!

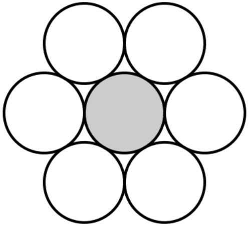
One more thing

Definition

The Kissing Number of $\mathbb{R}^n$ is the largest number of non-overlapping unit spheres which mutually "kiss" one central sphere.

e.g. When $n = 1$, the kissing number is 2. When $n = 2$, the kissing number is 6.





Open Question

What is the kissing number for $n \geq 5$?

It turns out this problem can be expressed as a first order formula over $(\mathbb{R}, +, \times, 0, 1, \leq)$.

As we now know, the theory of that structure is decidable! So we can ask a computer for the answer.

Why Think!

Thank You!