Countifiers

Kamal Lodaya
Joint work with *A V Sreejith*

50 years of IMSc

In the beginning

The axioms in Euclid's book *Elements*, from approximately 300 BCE, do not axiomatize Euclidean geometry.

- Nikolai Ivanovich Lobachevsky (1792-1856), in the period 1820-1826, published in *Vestnik Kazanskogo universiteta*, 1829-30
- János Bolyai (1802-1860), in the period 1820-1823, published by his father Farkas Bolyai (1775-1856) in 1832 as an appendix to his textbook
- Carl Friedrich Gauss (1777-1855), in the period 1820-1824, referred to in private correspondence but never published
- Gauss and Farkas Bolyai were in correspondence, Gauss praised his son's work.
- Lobachevsky never communicated with the others, Gauss read and praised his work.

A song by Tom Lehrer, 1953

Who made me the genius I am today,
The mathematician that others all quote,
Who's the professor that made me that way?
The greatest that ever got chalk on his coat.

One man deserves the credit,
One man deserves the blame,
And Nikolai Ivanovich Lobachevsky is his name.
Ha! Nikolai Ivanovich Lobach- . . .

I am never forget the day I first meet the great Lobachevsky.
In one word he told me secret of success in mathematics:
Plagiarize!

Plagiarize,
Let no one else's work evade your eyes,
Remember why the good Lord made your eyes,
So don't shade your eyes,
But plagiarize, plagiarize, plagiarize - . . .

What are the numbers and what should they be?

- Richard Dedekind (1831-1916) in his monograph *Was sind und was sollen die Zahlen?*, 1888
- Giuseppe Peano (1858-1932), in *Formulario Mathematico*, 1895, invited talk at the first ICM, Zürich, 1897
- Gave some axioms for arithmetic (notably induction)
- Gottlob Frege (1848-1925), over his books *Begriffsschrift*, 1879; *Die Grundlagen der Arithmetik*, 1884; *Grundgesetze der Arithmetik*, 1893 and 1903
- Defined predicate logic as a formal language

Predicate logic

Syntax

FO$[P_1, P_2, \dots]$ *with predicates* $P_1, P_2, \dots$ *and a countable set of variables*

$$\begin{aligned} \alpha \quad ::= \quad & P_i(x_1, \dots, x_{n_i}), i = 1, 2, \dots \mid x = y \quad \textit{atomic predicates} \\ & \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \alpha_1 \wedge \alpha_2 \quad \textit{boolean operations} \\ & \mid \exists x\ \alpha \mid \forall x\ \alpha \quad \textit{quantifiers} \end{aligned}$$

Given a domain D and an interpretation $P_i^D \subseteq D^{n_i}$:

- $(D, P_1^D, P_2^D, \dots) \models \exists x \alpha(x)$ iff for some $d \in D$, using the evaluation $[x \mapsto d]$, $(D, P_1^D, P_2^D, \dots) \models \alpha(.)$
- $(D, P_1^D, P_2^D, \dots) \models \forall x \alpha(x)$ iff for all $d \in D$, using the evaluation $[x \mapsto d]$, $(D, P_1^D, P_2^D, \dots) \models \alpha(.)$
- A formula is valid if it holds for all evaluations (and all interpretations, if they are allowed to vary)

$(\{barber, frege\}, \{(barber, frege), (barber, barber)\}) \models \exists x \forall y Shave(x, y)$ using the evaluation $[x \mapsto barber]$

A century ago

- Bertrand Russell (1872-1970) asked Frege in 1903: where do the domains come from? For example, if "*Shave*" is a binary predicate, and we talk about barbers who do not shave themselves, how do we define the domain ?
- Russell and Alfred North Whitehead (1861-1947) constructed in their book *Principia Mathematica*, 1910, the numbers axiomatically from a “type hierarchy” consisting of the empty set, sets containing other sets, sets containing sets containing other sets and so on.
- This separated first-order logic, where one quantifies over elements of the domain, from higher-order logics (also called *type theory*), where one can quantify over sets across the set hierarchy.

80 years ago

Theorem (Kurt Gödel, *Monatshefte für Mathematik und Physik*, 1931)

Validity of Peano arithmetic FO[+,×] interpreted over $\mathbb{N}$ ***cannot*** *be completely axiomatized.*

- In this paper Gödel (1906-1978) showed that self-referential sentences (like “proofs of formulae which do not prove the formulae themselves") can be coded in FO[+,×].
- In the formal definition of Peano arithmetic we have to use two ternary predicates $Add(x, y, z)$ and $Mult(x, y, z)$, but we will continue to use ordinary mathematical notation wherever possible.

Enter the computer

Theorem (Alan Turing, *Proceedings of the London Mathematical Society*, 1936-37)

*There is **no** algorithm*[1] *to check the validity of FO[+,×] when interpreted over* $\mathbb{N}$ *(or even initial segments of* $\mathbb{N}$*).*

- In this paper Turing (1912-1954) defined the machine which is now named after him[2] as a model of an algorithm, and showed that there is a universal Turing machine which can simulate all Turing machines.
- John von Neumann (1903-1957) used Turing's ideas for the architecture of a computer, *First draft of a report on the EDVAC*, 1945, with *hardware* (a UTM) executing *software* (programs like TMs). The EDVAC was built in 1949.

[1] A decision problem is undecidable if there is no algorithm to check it.

[2] Turing's theorem goes through for other definitions of machines, for example, "counter machines" or "register machines"

Weak fragments of addition and multiplication

Theorem (Raphael Robinson, *Proceedings of the American Mathematical Society*, 1958)

There is no algorithm to check validity of $FO[x+1, 2x, Red, Blue, \dots]$.

- Robinson (1911-1995) uses only a fragment of addition (successor) and a fragment of multiplication (doubling) but allows additional unary predicates[3].
- All the undecidability proofs use both addition and multiplication. What if we consider only one of them but not both?

[3] In computer science this is called an alphabet.

The dangers of writing a long tedious proof

Theorem (Mojżesz Presburger, *Comptes Rendus, Congrès des Mathématiciens des pays slaves*, 1929)

There is an algorithm to check the validity of an FO[+] formula.

Proof.

By quantifier elimination. Take an *innermost* formula $\exists x \alpha(x)$ (α has no quantifiers itself) and replace it by an *equivalent* (but exponentially larger) quantifier-free formula β. Work inside out. Finally, there is an algorithm to check the validity of a quantifier-free formula. □

- The adviser of Mr Presburger (1904-1943)[4] thought it wasn't worth a PhD and accepted it as a master's thesis.
- Today this is the most cited decision procedure in arithmetic.
- There is also a complete axiom system for FO[+].

[4] Alfred Tarski (1901-1983)

Other arithmetics

Theorem (Thoralf Skolem, *Skrifter Vitenskapsakadamiet i Oslo*, 1931)

There is an algorithm to check the validity of an FO[$\times$] formula.

Proof.

Five examples and the claim that they describe all cases. □

- Dr Skolem (1887-1963) had his degree by the time the paper was published.
- The first full proof was given by Andrzej Mostowski (1913-1975) in *Journal of Symbolic Logic*, 1952.
- A complete axiom system for FO[$\times$] was given by Patrick Cégielski, *Comptes Rendus*, 1980.

So FO[+] (logic of integers as an ordered group with addition) and FO[$\times$] (logic of positive numbers as an ordered group with multiplication) have nice properties, putting them together in FO[+,$\times$] (logic of integers as a ring) makes it go ballistic.

Counting with quantifiers

Now onwards we confine ourselves to initial segments of $\mathbb{N}$. For example $FO[<, P_1, P_2, \dots]$ where $<$ is interpreted as a finite linear order.

- $FO[<, Red]$ cannot express the property that the number of red points is even.
- Neither can $FO[<, +, \times, Red]$.
- Neither can $FO[<, P_1, P_2, \dots, Red]$ where the P_i are interpreted as any kind of numerical predicate over $\mathbb{N}$.
- By saying that the last position x is even (which is certainly a numerical predicate), we can say that the *total* number of points is even. But not the number of red points.

Countifier logic

Syntax

*FO*COUNT$[<, P_1, P_2, \dots]$ *with predicates* $P_1, P_2, \dots$ *and a countable set of variables*

$$\begin{aligned} \alpha \quad ::= \quad & P_i(x_1, \dots, x_{n_i}), i = 1, 2, \dots \mid x = y \mid x < y \mid \\ & \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \alpha_1 \wedge \alpha_2 \mid \\ & x = \#y\ \alpha \quad \textit{countifier} \end{aligned}$$

Given a *finite linear order* $(D, <)$ and an interpretation $P_i^D \subseteq D^{n_i}$ for the predicates,
$(D, P_1^D, P_2^D, \dots) \models x = \#y\alpha(y)$ iff the value of x is the count $\#\{d \mid$ using the evaluation $[y \mapsto d]$, $(D, P_1^D, P_2^D, \dots) \models \alpha(.)\}$

- $\exists y\alpha(y)$ is definable as $x = \#y\alpha(y) \wedge x > 0$
- $\forall y\alpha(y)$ is definable as $x = \#y\neg\alpha(y) \wedge x = 0$
- $(x = \#y\ Red(y)) \wedge \exists y(x = y + y)$ defines an even number of red points

Counting can say a lot

Just addition: Divisibility by a constant $k|x$ is definable as $\exists q(q + q + \cdots + q = x)$

FOCOUNT over a linear order: Addition $x + y = z$ is definable as $y = \#b(x \leq b \wedge b < z)$, more cryptically as $y = \#y(x \leq y \wedge y < z)$ without using an extra variable, so FOCOUNT[$<$] and FOCOUNT[+] are the same.

Multiplication: Divisibility $x|y$ is of course definable as $\exists z(x \times z = y)$,
but then also "x is prime" is definable as
$x > 1 \wedge \forall q(q|x \Rightarrow q = 1 \vee q = x)$

FOCOUNT with multiplication: Define $\#x(\mathit{prime}(x) \wedge x < n)$ and formulate the prime number theorem.

So one can express more with counting quantifiers. What is the price one has to pay?

Two-variable counting is already hard

Syntax

FO^k stands for first-order logic with k variables (instead of countably many in FO).

Theorem (Erich Grädel, Martin Otto and Eric Rosen, Stacs 1997)

There is no algorithm to check validity of $FO^2\mathrm{COUNT}[<, x+1, Red, Blue, Green, Yellow]$.

Proof.

Let *Red* stand for "increment" and *Blue* for "decrement", *Green* and *Yellow* the same with another counter. Now simulate the run of a two-counter machine. The formula

$$y = \#y(y < x \wedge Red(y)) \wedge y = \#y(y < x \wedge Blue(y))$$

is true at point x only if the value of the first counter is zero. □

Two-variable addition is bad enough

Theorem

There is no algorithm to check validity of $FO^2[x+1, 2x, Red, Blue, \ldots]$.

Proof.

- Simulate in this restricted logic checking the emptiness of a halting Turing machine which uses space n. (Since it halts, there must be such an n.)
- Let positions $n+1, n+2, \ldots, 2n$ contain the initial configuration of the Turing machine (coded using colours).
- Let positions $2n+2, 2n+4, \ldots, 4n$ have the next configuration, the inbetween positions are filled up with suitable junk.
- Let positions $4n+4, 4n+8, \ldots, 8n$ have the next configuration.
- The halting configuration is marked by a special colour.

On two-variable addition

- In the last proof, we used the predicates:
  $Successor(x, y) \stackrel{\text{def}}{=} y = x + 1$ and
  $Double(x, y) \stackrel{\text{def}}{=} y = x + x$
- Both of these are definable from addition using just two variables. That is, instead of using a ternary addition predicate $Add(x, y, z)$ we restricted addition to whatever can be done using two variables.
- So the proof shows that when additional unary predicates (like $Red, Blue, \ldots$) are allowed, even two-variable Presburger arithmetic is undecidable, a strengthening of Raphael Robinson's result of 1958.
- Now onwards we will also disallow these additional unary predicates[5] and consider a single ternary predicate $Add(x, y, z)$ interpreted over initial segments of $\mathbb{N}$.

[5] Alternately the alphabet is of size 1

Countifiers with addition

Theorem (Nicole Schweikardt, *ACM Transactions on computational logic*, 2005)

*There is an algorithm to translate an FO*COUNT*[+] formula over* $\mathbb{N}$ *to an FO[+] formula.*

Proof.

Using back-and-forth games devised by Andrzej Ehrenfeucht in *Fundamenta Mathematicae*, 1961. Earlier work by Roland Fraïssé (1920-2008), *Comptes Rendus*, 1950; *Publications Scientifiques de l'Université d'Alger*, 1954. □

- So validity of FOCOUNT[+] is checkable.
- Presburger, Skolem and Schweikardt's algorithms have nonelementary complexity. Can we do better?

David Cooper, in *Machine Intelligence*, 1972

Consider the formula with the constraints
$\exists x(2x < 2y+3z+5w,\ 3y+5z+2w < 3x,\ 5x < 5y+2z+3w)$.

- Here negations have been driven in as far as possible, equalities have been taken out, negations of equalities, replaced by inequalities, and only $<$ and modulo constraints remain.
- The *lcm* of the coefficients of x is 30. Rewrite the formula to make the coefficient of x equal to 30 everywhere and then replace $30x$ by x':

$$\begin{aligned}\exists x'(\quad & x' \equiv 0 \mod 30,\\ & x' < 30y + 45z + 75w,\ x' < 30y + 12z + 18w,\\ & 30y + 50z + 20w < x')\end{aligned}$$

- The *lcm* of the modulo quotients is 30. Cooper's theorem eliminates the quantifier and rewrites the formula to:

$$\begin{aligned}\textstyle\bigvee_{j=1}^{30}(\quad & 30y + 50z + 20w + j \equiv 0 \mod 30,\\ & 30y + 50z + 20w + j < 30y + 45z + 75w,\\ & 30y + 50z + 20w + j < 30y + 12z + 18w)\end{aligned}$$

A chase from 40 years ago

- Cooper's algorithm to check validity of FO[+] searches for solutions separately among the congruence classes and avoids the repeated conversions to disjunctive normal form (DNF) in Presburger's algorithm.
- Derek Oppen (STOC 1973) showed that the complexity of Cooper's algorithm is 3EXPTIME.
- Michael Fischer and Michael Rabin (SIAM-AMS Symposium on Complexity of Computations, 1974) showed that there is a lower bound of 2EXPTIME for checking the validity of FO[+] formulae over $\mathbb{N}$.
- Jeanne Ferrante and Charles Rackoff (*SIAM Journal on Computing*, 1975) gave an upper bound of 2EXPSPACE.
- Leonard Berman (FOCS 1977) showed lower and upper bounds of ATIME$[2^{2^{O(n)}}, O(n)]$[6], alternating double exponential time with a linear number of alternations.

[6] 2EXPTIME$\subseteq$ ATIME$[2^{2^{O(n)}}, O(n)]\subseteq$ 2EXPSPACE$\subseteq$ 3EXPTIME

Revisiting Cooper for countifier elimination

Suppose now that n is to count solutions to the constraints $\#x(2x < 2y+3z+5w,\ 3y+5z+2w < 3x,\ 5x < 5y+2z+3w)$.

- By following Cooper's algorithm, we get:

$$\begin{aligned} n = \#x'(\quad & x' \equiv 0 \mod 30, \\ & x' < 30y + 12z + 18w,\ x' < 30y + 12z + 18w, \\ & 30y + 50z + 20w < x') \end{aligned}$$

- To generalize Cooper's theorem to counting, we have to choose one of the terms to be a greatest lower bound for x' and one of the terms to be a least upper bound.

$$\begin{aligned} \bigvee_{j=1}^{30} \quad [\quad & 30y + 50z + 20w + j \equiv 0 \mod 30, \\ (\quad & 30y + 50z + 20w + j < 30y + 12z + 18w, \\ & 30y + 12z + 18w < 30y + 45z + 75w, \\ & n = \lfloor (-38z - 2w - 1)/30 \rfloor) \\ \vee \quad & (30y + 50z + 20w + j < 30y + 45z + 75w, \\ & 30y + 45z + 75w \leq 30y + 12z + 18w, \\ & n = \lfloor (-5z - 55w - 1)/30 \rfloor)] \end{aligned}$$

Complexity questions

Considering all the cases is a bit involved, but we think Oppen's technique works.

Conjecture

*The validity of FO*COUNT*[+] is checkable in elementary time.*

We are not sure of the exact complexity, we think it is 3EXPTIME[7].

Question

Can we confirm this? Can we do better?

[7] Recall ATIME$[2^{2^{O(n)}}, O(n)] \subseteq$ 2EXPSPACE $\subseteq$ 3EXPTIME

A weaker logic

Syntax

*FO*MOD *allows only the restricted comparison* $\#y\alpha(y) \equiv r \mod q$ *(that is, count the* α*'s and immediately compare, instead of the general syntax* $x = \#y\alpha(y)$ *in* *FO*COUNT *which allows storing the count in a variable* x*).*

- This is sufficient to add some of the counting capability lacking in first-order logic.
- $\#yRed(y) \equiv 0 \mod 2$: an even number of red points.

Theorem

*The validity of FO*MOD*[+] is in* 2EXPSPACE*.*

Proof.

Ferrante and Rackoff's idea of refining an Ehrenfeucht-Fraïssé game. □

Question

Can we do better?

In conclusion

Moral Countifiers are expressive and at the level of Presburger arithmetic, their price doesn't seem to be too high.

Aside Counting complexity is an important area of computer science. For example, we do not know if solving a counting problem in polynomial time has the same complexity as solving a decision problem in polynomial time (more precisely, FPTIME= #PTIME).

Side story Hope you enjoyed the history of how mathematicians communicate their results to others.

Thank you for your attention.