Decision Problems in Algebra

(FCUL Summer School)

António Malheiro

CMA/FCT
Universidade Nova de Lisboa

Acknowledgement:
This work was supported by CMA within the projects
UID/MAT/00297/2013
PTDC/MHC-FIL/2583/2014
financed by 'Fundação para a Ciência e a Tecnologia'

June 2018

Outline

Fundamental Dehn's Decision Problems

Undecidability

- H10;
- Turing Machines;
- Recursively enumerable and recursive sets;
- The Halting problem;
- Undecidability of the word problem;
- Markov properties.

Related topics

H10

Do there exist integers x, y, z such that

$$x^3 + y^3 + z^3 = 29?$$

Yes: $(x, y, z) = (3, 1, 1)$.

$$x^3 + y^3 + z^3 = 30?$$

Yes: $(x, y, z) = (-283059965, -2218888517, 2220422932)$.

$$x^3 + y^3 + z^3 = 33?$$

Unknown.

H10

David Hilbert

Hilbert's tenth problem (H10)

Find an algorithm that solves the following problem:

input: a multivariable polynomial $f(x_1, \ldots, x_n)$ with integer coefficients;

output: YES or NO, according to whether there exist integers $a_1, a_2, \ldots, a_n$ such that $f(a_1, \ldots, a_n) = 0$.

H10

Theorem (Davis–Putnam–Robinson 1961 + Matiyasevich 1970)

No such algorithm exists!

To be precise we need to know ...

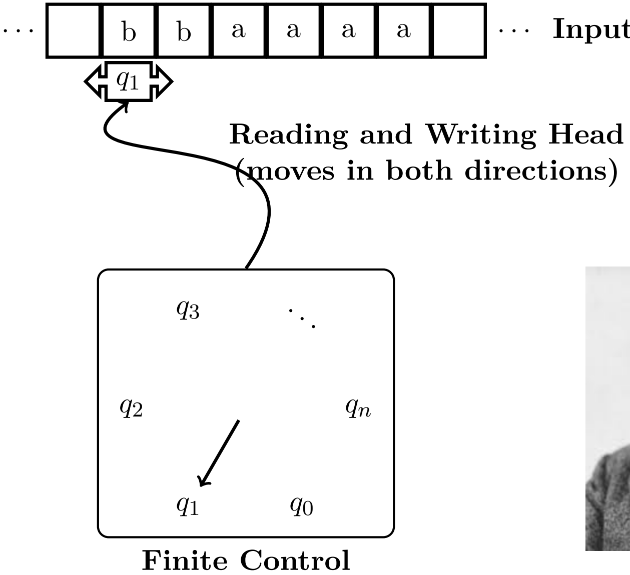
What is an algorithm?

Such notions were introduced in the 1930s by ...

Alonzo Church

Alan Turing

Turing Machines



Alan Turing

The Church–Turing thesis

Any finite written description of a deterministic step-by-step computation is equivalent to some Turing machine.

Moreover, there is a construction that, given such a finite description, will give us an explicit Turing machine T that carries out the original computation.

Turing Machines

- T a Turing Machine
- w a word in the alphabet A
- $T(w)$ the computation of the TM on input w
- $T(w)$ halts if the computation $T(w)$ halts eventually in finitely many steps - write $T(w)\downarrow$
- if T never reaches its halting state on input w, then we say $T(w)$ does not halt, and write $T(w)\uparrow$
- the language recognized by T is $\Omega(T) = \{w \in A^+ \mid T(w)\downarrow\}$

Recursively enumerable

A subset S of A^+ is called recursively enumerable if $S = \Omega(T)$ for some Turing machine T.

Recursive

A subset S of A^+ is called recursive if both S and $A^+ \backslash S$ are recursively enumerable.

Turing Machines

Any Turing machine T on (ordered) alphabet A defines a partial function $f : \mathbb{N} \to \mathbb{N}$.

Every Turing machine can be encoded as a natural number.

Theorem (Turing 1937)

There exists a universal Turing machine. That is, a Turing machine which can simulate the action of every Turing machine.

Sketch of the proof:

Define an algorithm T as follows: on input of an integer $x = \langle m, n \rangle$; compute $T_m(n)$.
So T is such that $T(x) = T_m(n)$ (that is, if $T_m(n) \downarrow$ with output k then $T(x) \downarrow$ with output k, and if $T_m(n) \uparrow$ then $T(x) \uparrow$).

The halting problem

The Halting Problem

input: a Turing machine T and an input w;

output: YES or NO, according to whether T halts on w or not.

Definition (The Halting set)

$$\mathbb{K} = \{n \in \mathbb{N} \mid T_n(n) \downarrow\}.$$

Theorem

The halting set $\mathbb{K}$ satisfies the following:

1. $\mathbb{K}$ is recursively enumerable;
2. $\mathbb{K}$ is not recursive.

The halting problem

Proof (sketch):

(1) In the same way that we defined a universal Turing machine $T(x) := T_m(n)$ (where $x = \langle m, n \rangle$), we can define a 'restricted' universal Turing machine that computes one entry of each Turing machine, by $T'(n) := T_n(n)$.

(2) Suppose that $\mathbb{K}$ was recursive. So there exists some TM, $m := T_m$, such that $\Omega(m) = \mathbb{N} \backslash \mathbb{K}$. We have:

$$\begin{aligned} m \in \mathbb{K} &\Leftrightarrow T_m(m)\downarrow \\ &\Leftrightarrow m \in \Omega(m) \\ &\Leftrightarrow m \in \mathbb{N} \backslash \mathbb{K} \\ &\Leftrightarrow m \notin \mathbb{K}. \end{aligned}$$

Contradiction.

Back to H10

Definition (Diophantine set)

$A \subseteq \mathbb{Z}$ is called diophantine if there exists $p(t, \vec{x}) \in \mathbb{Z}[t, x_1, \ldots, x_m]$ such that

$$A = \{a \in \mathbb{Z} \mid \ (\exists \vec{x} \in \mathbb{Z}^m)\ p(a, \vec{x}) = 0\}.$$

Example

The subset $\mathbb{N}$ of $\mathbb{Z}$ is diophantine, since for $a \in \mathbb{Z}$,

$$a \in \mathbb{N} \ \Leftrightarrow \ (\exists x_1, \ldots, x_4 \in \mathbb{Z})\ x_1^2 + x_2^2 + x_3^2 + x_4^2 = a.$$

Theorem (Davis–Putnam–Robinson & Matiyasevich)

Diphantine $\Leftrightarrow$ Recursively enumerable

- The unsolvability of the halting problem provides a recursively enumerable set for which no algorithm can decide membership.
- So there exists a diophantine set for which no algorithm can decide membership.

Undecidability of the word problem

Given a f.p. group G, we have

Word problem for G:

input: a word w in the generators of G

output: YES or NO, according to whether w represents the identity in G.

Theorem (Novikov & Boone (independently) 1950's)

There exists a f.p. group G such that the word problem for G is undecidable.

Proof strategy:

Construct a group G for which solve the word problem is at least as hard as solving the halting problem.

Corollary

The uniform word problem is undecidable.

Markov properties

Definition

A property of f.p. groups is said to be a Markov property if:

- there exists a f.p. group G with the property; and
- there exists a f.p. group H that cannot be embedded in any f.p. group with the property.

Example (of Markov properties)

- being finite;
- trivial;
- abelian;
- free;
- ...

Markov properties

Theorem (Adian & Rabin 1955-1958)

For each Markov property $\mathcal{P}$, the problem of deciding whether an arbitrary f.p. group has $\mathcal{P}$ is undecidable.

Sketch of proof:

Embed the uniform word problem in this $\mathcal{P}$ problem: Given an f.p. group G and a word w in its generators, build another f.p. group K such that

$$K \text{ has } \mathcal{P} \Leftrightarrow w = 1 \text{ in } G.$$