The Legacy of Turing in Numerical Analysis

Felipe Cucker

City University of Hong Kong

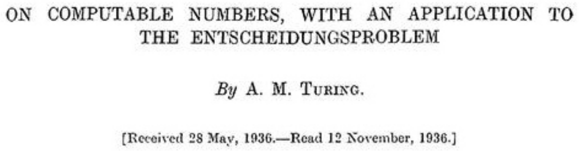
SOFSEM
January 2012

How do people know Alan Mathison Turing?

- The layman: mostly because of his role in decyphering the *enigma* code during WWII.

How do people know Alan Mathison Turing?

- The layman: mostly because of his role in decyphering the *enigma* code during WWII.
- The (computer?) scientist: mostly as the author of



ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

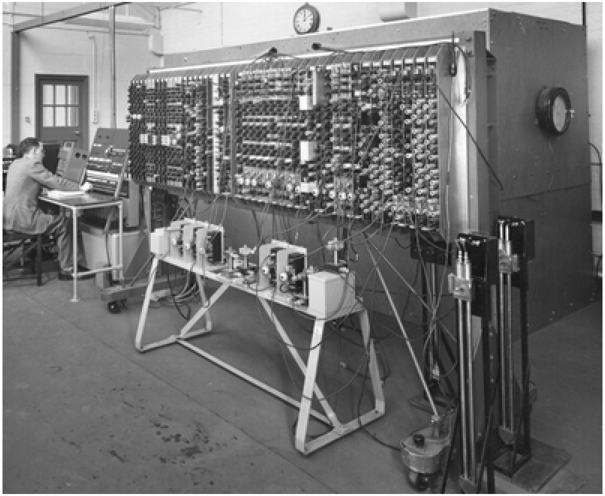
where the Turing machine is introduced.

The Turing machine became central in the theoretical foundations of computer science because, unlike other abstract models of computation leading to the same class of computable functions, the Turing machine provides a natural framework to define, and subsequently study, issues of complexity.

The development of computer languages, compilers, editors, web browsers, &c, brought to light a myriad of combinatorial problems demanding efficient algorithmic solutions and the complexity theory built upon the Turing machine has been pivotal in the phenomenal progress this development has seen.

But there were other aspects in Turing's life . . .

and other aspects in Turing's work . . .



Between 1946 and 1948 Turing worked in the National Physical Laboratory (as a numerical analyst).

> *Turing's international reputation rests mainly on his work on computable numbers but I like to recall that he was a considerable numerical analyst, and a good part of his time from 1946 onwards was spent working in this field [...]*
>
> James Wilkinson, *1970's Turing Lecture.*

Theoretical Issues in Numerical Analysis: Complexity and Accuracy

Since none of the numbers we take out from logarithmic or trigonometric tables admit of absolute precision, but are all to a certain extent approximate only, the results of all calculations performed by the aid of these numbers can only be approximately true. [. . .] It may happen, that in special cases the effect of the errors of the tables is so augmented that we may be oblidged to reject a method, otherwise the best, and substitute another in its place.

Carl Friedrich Gauss, *Theoria Motus.*

Implicit in Gauss' words we have that:

- Real number computations are polluted by errors and these errors accumulate during the computation.

Implicit in Gauss' words we have that:

- Real number computations are polluted by errors and these errors accumulate during the computation.
- If the resulting final error is too large we may be forced to use another algorithmic solution even if the original one was *otherwise the best*.

Implicit in Gauss' words we have that:

- Real number computations are polluted by errors and these errors accumulate during the computation.
- If the resulting final error is too large we may be forced to use another algorithmic solution even if the original one was *otherwise the best*.
- Algorithmic design involves two considerations:

Implicit in Gauss' words we have that:

- Real number computations are polluted by errors and these errors accumulate during the computation.
- If the resulting final error is too large we may be forced to use another algorithmic solution even if the original one was *otherwise the best*.
- Algorithmic design involves two considerations:
    - Complexity (find most efficient, i.e. fastest, algorithm): one assumes infinite precision.

Implicit in Gauss' words we have that:

- Real number computations are polluted by errors and these errors accumulate during the computation.
- If the resulting final error is too large we may be forced to use another algorithmic solution even if the original one was *otherwise the best*.
- Algorithmic design involves two considerations:
  - Complexity (find most efficient, i.e. fastest, algorithm): one assumes infinite precision.
  - Accuracy (estimate accumulation of errors): one assumes finite precision.

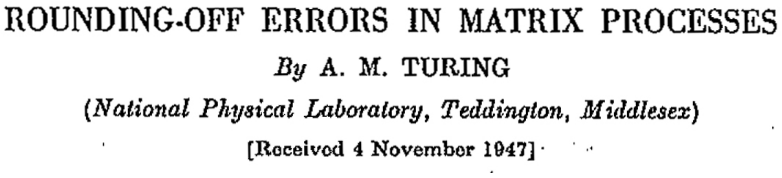
(1) Accuracy

According to Wilkinson,

> *some time after my arrival, a system of 18 equations arrived in Mathematics Division and after talking around it for some time we finally decided to abandon theorizing and to solve it. [. . .] we decided on Gaussian elimination with complete pivoting. Turing was not particularly enthusiastic, partly because he was not an experienced performer on a desk machine and partly because he was convinced that it would be a failure.*

Instead, the computation was a total success and it was this success (against Turing's initial apathy) that

> *set him thinking afresh on the problem of rounding errors in elimination processes. About a year later he produced his famous paper [. . .]*



ROUNDING-OFF ERRORS IN MATRIX PROCESSES

By A. M. TURING

(*National Physical Laboratory, Teddington, Middlesex*)

[Received 4 November 1947]

A finite-precision algorithm working with a *machine precision* ϵ_{mach}, $0 < \epsilon_{\mathsf{mach}} < 1$, replaces, during the computation, all numbers x by a number $\tilde{x}$ such that

$$\tilde{x} = x(1 - \delta) \text{ with } |\delta| \leq \epsilon_{\mathsf{mach}}.$$

A finite-precision algorithm working with a *machine precision* ϵ_{mach}, $0 < \epsilon_{\mathrm{mach}} < 1$, replaces, during the computation, all numbers x by a number $\tilde{x}$ such that

$$\tilde{x} = x(1 - \delta) \text{ with } |\delta| \leq \epsilon_{\mathrm{mach}}.$$

If $a \in \mathbb{R}^n$ is approximated by $\tilde{a}$ we may define the *relative error* of this approximation by taking

$$\mathrm{RelError}(a) = \frac{\|a - \tilde{a}\|}{\|a\|}.$$

A finite-precision algorithm working with a *machine precision* ϵ_{mach}, $0 < \epsilon_{\text{mach}} < 1$, replaces, during the computation, all numbers x by a number $\tilde{x}$ such that

$$\tilde{x} = x(1-\delta) \text{ with } |\delta| \leq \epsilon_{\text{mach}}.$$

If $a \in \mathbb{R}^n$ is approximated by $\tilde{a}$ we may define the *relative error* of this approximation by taking

$$\mathrm{RelError}(a) = \frac{\|a-\tilde{a}\|}{\|a\|}.$$

Now assume we have a function $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ and an algorithm $\mathscr{A}$ meant to compute it. That is, $\mathscr{A}$ actually computes a function $\varphi^{\mathscr{A}}$ which depends on ϵ_{mach} and which coincides with φ under infinite precision ($\epsilon_{\text{mach}} = 0$). The central question underlying Gauss' quote can be rephrased as follows:

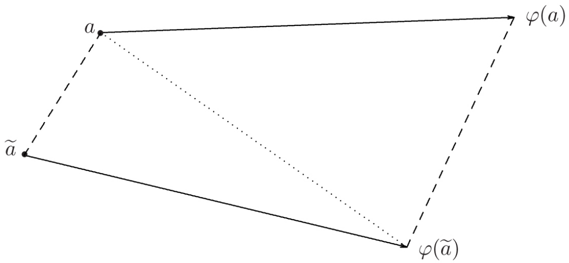
How big is $\mathrm{RelError}(\varphi^{\mathscr{A}}(a))$?

The answer to this question relies on two ingredients:

- the nature of the algorithm $\mathscr{A}$
- a magnification factor depending solely on a and φ.

We can define this magnification factor by taking the worst-case magnification in $\varphi(a)$ of small errors in a:

$$\mathrm{cond}^{\varphi}(a) := \lim_{\delta\to 0} \sup_{\mathrm{RelError}(a)\leq\delta} \frac{\mathrm{RelError}(\varphi(a))}{\mathrm{RelError}(a)}.$$



Let A be a square matrix.

$$\varphi(A) := A^{-1} \quad \Longrightarrow \quad \mathrm{cond}^{\varphi}(A) = \|A\|\|A^{-1}\|.$$

$$\varphi(A, b) := A^{-1}b \quad \Longrightarrow \quad \|A\|\|A^{-1}\| \leq \mathrm{cond}^{\varphi}(A, b) \leq 2\|A\|\|A^{-1}\|.$$

Let A be a square matrix.

$$\varphi(A) := A^{-1} \quad \Longrightarrow \quad \mathrm{cond}^{\varphi}(A) = \|A\|\|A^{-1}\|.$$
$$\varphi(A,b) := A^{-1}b \quad \Longrightarrow \quad \|A\|\|A^{-1}\| \leq \mathrm{cond}^{\varphi}(A,b) \leq 2\|A\|\|A^{-1}\|.$$

It follows that

$$\kappa(A) := \|A\|\|A^{-1}\|$$

measures, maybe up to a small factor, the worst case magnification in A^{-1} or in $x = A^{-1}b$ of small errors in the input A (resp. (A, b)). This quantity is known as the *condition number* of A.

Let A be a square matrix.

$$\varphi(A) := A^{-1} \quad \Longrightarrow \quad \mathrm{cond}^{\varphi}(A) = \|A\|\|A^{-1}\|.$$
$$\varphi(A, b) := A^{-1}b \quad \Longrightarrow \quad \|A\|\|A^{-1}\| \leq \mathrm{cond}^{\varphi}(A, b) \leq 2\|A\|\|A^{-1}\|.$$

It follows that

$$\kappa(A) := \|A\|\|A^{-1}\|$$

measures, maybe up to a small factor, the worst case magnification in A^{-1} or in $x = A^{-1}b$ of small errors in the input A (resp. (A, b)). This quantity is known as the *condition number* of A.

Both $\kappa(A)$ and its name appear in Turing's 1948 paper.

Section 8 of Turing's paper carries the title "Ill-conditioned matrices and equations" and starts with the following statement:

> *When we come to make estimates of errors in matrix processes we shall find that the chief factor limiting the accuracy that can be obtained is 'ill-conditioning' of the matrices involved*

Section 8 of Turing's paper carries the title "Ill-conditioned matrices and equations" and starts with the following statement:

> *When we come to make estimates of errors in matrix processes we shall find that the chief factor limiting the accuracy that can be obtained is 'ill-conditioning' of the matrices involved*

and continues

> *It is characteristic of ill-conditioned sets of equations that small percentage errors in the coefficients given may lead to large percentage errors in the solution.*

Section 8 of Turing's paper carries the title "Ill-conditioned matrices and equations" and starts with the following statement:

> *When we come to make estimates of errors in matrix processes we shall find that the chief factor limiting the accuracy that can be obtained is 'ill-conditioning' of the matrices involved*

and continues

> *It is characteristic of ill-conditioned sets of equations that small percentage errors in the coefficients given may lead to large percentage errors in the solution.*

Turing then defined $\kappa(A)$ "as a measure of the degree of ill-conditioning in a matrix" and proceeded to derive error estimates for some algorithmic methods.

Further advances

Independently of Turing, John von Neumann and Herman Goldstine were pursuing similar thoughts. They introduced a theme which, subsequently championed by Steve Smale, would become central in the foundations of numerical analysis: the probabilistic analysis of condition numbers.

Further advances

Independently of Turing, John von Neumann and Herman Goldstine were pursuing similar thoughts. They introduced a theme which, subsequently championed by Steve Smale, would become central in the foundations of numerical analysis: the probabilistic analysis of condition numbers.

The motivation is clear. Error bounds (and some complexity bounds as well) depend on the condition $\mathrm{cond}^{\varphi}(a)$ of the input a. But we do not know this quantity and it has been observed that its computation requires, essentially, to compute $\varphi(a)$.

Further advances

Independently of Turing, John von Neumann and Herman Goldstine were pursuing similar thoughts. They introduced a theme which, subsequently championed by Steve Smale, would become central in the foundations of numerical analysis: the probabilistic analysis of condition numbers.

The motivation is clear. Error bounds (and some complexity bounds as well) depend on the condition $\mathrm{cond}^{\varphi}(a)$ of the input a. But we do not know this quantity and it has been observed that its computation requires, essentially, to compute $\varphi(a)$.

A way out from this vicious circle is to estimate the expectation of $\mathrm{cond}^{\varphi}(a)$ (or, more commonly, of its logarithm) for random a. Goldstine and von Neumann did not go that far. But Alan Edelman did proving, for random $n \times n$ matrices (with independent Gaussian entries), that

$$\mathbb{E}(\log \kappa(A)) = \log n + C + o(1), \qquad \text{as } n \to \infty,$$

with $C = 1.537$ for real matrices and $C = 0.982$ for complex matrices.

Further advances

Independently of Turing, John von Neumann and Herman Goldstine were pursuing similar thoughts. They introduced a theme which, subsequently championed by Steve Smale, would become central in the foundations of numerical analysis: the probabilistic analysis of condition numbers.

The motivation is clear. Error bounds (and some complexity bounds as well) depend on the condition $\mathrm{cond}^{\varphi}(a)$ of the input a. But we do not know this quantity and it has been observed that its computation requires, essentially, to compute $\varphi(a)$.

A way out from this vicious circle is to estimate the expectation of $\mathrm{cond}^{\varphi}(a)$ (or, more commonly, of its logarithm) for random a. Goldstine and von Neumann did not go that far. But Alan Edelman did proving, for random $n \times n$ matrices (with independent Gaussian entries), that

$$\mathbb{E}(\log \kappa(A)) = \log n + C + o(1), \qquad \text{as } n \to \infty,$$

with $C = 1.537$ for real matrices and $C = 0.982$ for complex matrices. This result produces upper bounds on the expected loss of precision for matrix inversion and linear equation solving.

(2) Complexity

After the overview on Turing's contribution to the second stage in Gauss' scheme, we can start considering the same for the first.

Here Turing's role is bigger and with more disparate origins. It is not controversial today to state that his 1936 paper is at the center of contemporary complexity theory. Two features of the paper stand out:

- the introduction of the Turing machine, which allows for a formal definition of the *cost* of a computation.
- the use of *reductions* between computational problems as a means to state that one problem requires no less resources (i.e., cost) than another for its algorithmic solution.

(2) Complexity

After the overview on Turing's contribution to the second stage in Gauss' scheme, we can start considering the same for the first.

Here Turing's role is bigger and with more disparate origins. It is not controversial today to state that his 1936 paper is at the center of contemporary complexity theory. Two features of the paper stand out:

- the introduction of the Turing machine, which allows for a formal definition of the *cost* of a computation.
- the use of *reductions* between computational problems as a means to state that one problem requires no less resources (i.e., cost) than another for its algorithmic solution.

Turing introduced reductions with the goal of classifying undecidable problems but his ideas eventually translated into a classification of decidable problems (in terms of resource requirements). Thus, for instance, the class P is that of all sets decidable within a cost polynomial in the size of the input. Similarly, the class NP is that of the sets decidable within a non-deterministic polynomial cost.

Problems that can be proved to be in NP $\setminus$ P are considered to be intrinsically difficult to solve and the best candidates in this class are the so called NP-*complete* problems, defined in terms of a variation on the reductions introduced by Turing. The first example of such a problem was independently exhibited by Steve Cook and Leonid Levin. Shortly after, Richard Karp proved NP-completeness for 23 problems in different areas of computation showing that the phenomenon was much more general than Cook and Levin's results suggested and hundreds of problems have joined the list since then.

Problems that can be proved to be in NP $\setminus$ P are considered to be intrinsically difficult to solve and the best candidates in this class are the so called NP-*complete* problems, defined in terms of a variation on the reductions introduced by Turing. The first example of such a problem was independently exhibited by Steve Cook and Leonid Levin. Shortly after, Richard Karp proved NP-completeness for 23 problems in different areas of computation showing that the phenomenon was much more general than Cook and Levin's results suggested and hundreds of problems have joined the list since then.

These considerations apply to *discrete computations*. Basically, these are computations whose intervening objects can be precisely described with a finite word over a finite alphabet. In opposition to these computations are those of *continuous mathematics*, that is, those involving real or complex numbers, commonly known as *numerical algorithms*. These can only be approximated by finite words over a finite alphabet (and hence Gauss' tribulations on the accumulation of errors).

Problems that can be proved to be in $\mathrm{NP} \setminus \mathrm{P}$ are considered to be intrinsically difficult to solve and the best candidates in this class are the so called NP-*complete* problems, defined in terms of a variation on the reductions introduced by Turing. The first example of such a problem was independently exhibited by Steve Cook and Leonid Levin. Shortly after, Richard Karp proved NP-completeness for 23 problems in different areas of computation showing that the phenomenon was much more general than Cook and Levin's results suggested and hundreds of problems have joined the list since then.

These considerations apply to *discrete computations*. Basically, these are computations whose intervening objects can be precisely described with a finite word over a finite alphabet. In opposition to these computations are those of *continuous mathematics*, that is, those involving real or complex numbers, commonly known as *numerical algorithms*. These can only be approximated by finite words over a finite alphabet (and hence Gauss' tribulations on the accumulation of errors).

To discuss complexity over $\mathbb{R}$ we return to Turing.

It is not surprising that Turing's 1948 paper begins with a short section on cost. Indeed, its Section 1, bearing the title "Measure of work in a process", can be quoted in its entirety:

> *It is convenient to have a measure of the amount of work involved in a computing process, even though it be a very crude one. We may count up the number of times that various elementary operations are applied in the whole process and then give them various weights. We might, for instance, count the number of additions, subtractions, multiplications, divisions, recordings of numbers, and extractions of figures from tables. In the case of computing with matrices most of the work consists of multiplications and writing down numbers, and we shall therefore only attempt to count the number of multiplications and recordings. For this purpose, a reciprocation will count as a multiplication. This is purely formal. A division will then count as two multiplications; this seems a little too much, and there may be other anomalies, but on the whole substantial justice should be done.*

That is, Turing is disregarding a measure of cost in terms of elementary bit operations (as would correspond to a Turing machine!) and is suggesting instead what is usually called *algebraic cost*, in which floating-point numbers are considered as undivisible entities and arithmetic operations between them as elementary machine operations.

That is, Turing is disregarding a measure of cost in terms of elementary bit operations (as would correspond to a Turing machine!) and is suggesting instead what is usually called *algebraic cost*, in which floating-point numbers are considered as undivisible entities and arithmetic operations between them as elementary machine operations.

This pondering was not meant to build a complexity theory as the one underlying the P vs NP problem. It barely aimed to provide a framework within which an idea of cost for numerical algorithms can be agreed on and algorithms can therefore being compared w.r.t. cost as implicit in Gauss' quotation.

For discrete computations, the definition of the class P did not arrive until the 60s and the structural approach to complexity until the early 70s (with the papers of Cook, Levin, and Karp).

For discrete computations, the definition of the class P did not arrive until the 60s and the structural approach to complexity until the early 70s (with the papers of Cook, Levin, and Karp).

For numerical computations, it is not until the late 80s that a paper by Lenore Blum, Michael Shub, and Steve Smale would put together the two streams of thought initiated by Turing we mentioned above [36 & 48]. Blum, Shub and Smale define an abstract model of computation, which they call *real Turing machine* (but has since been refered to as *BSS-machine*), and associate to it a notion of cost which essentially coincides with that of Section 1 in Turing's paper. Furthermore, they extended the notion of nondeterminism, defined the class $\mathrm{NP}_{\mathbb{R}}$ of sets decidable in nondeterministic polynomial time and exhibited an $\mathrm{NP}_{\mathbb{R}}$-complete problem (a task for which, Turing's idea of reduction was again of the essence).

For discrete computations, the definition of the class P did not arrive until the 60s and the structural approach to complexity until the early 70s (with the papers of Cook, Levin, and Karp).

For numerical computations, it is not until the late 80s that a paper by Lenore Blum, Michael Shub, and Steve Smale would put together the two streams of thought initiated by Turing we mentioned above [36 & 48]. Blum, Shub and Smale define an abstract model of computation, which they call *real Turing machine* (but has since been refered to as *BSS-machine*), and associate to it a notion of cost which essentially coincides with that of Section 1 in Turing's paper. Furthermore, they extended the notion of nondeterminism, defined the class $\mathrm{NP}_{\mathbb{R}}$ of sets decidable in nondeterministic polynomial time and exhibited an $\mathrm{NP}_{\mathbb{R}}$-complete problem (a task for which, Turing's idea of reduction was again of the essence).

In the rest of this talk I will give some details of this complexity theory over the reals.

We denote by $\mathbb{R}^\infty$ the disjoint union $\bigsqcup_{n\geq 1}\mathbb{R}^n$. This is the space where inputs are taken from in numerical computations and, in fact, a BSS-machine M computes a (partial) function $\varphi^M : \mathbb{R}^\infty \to \mathbb{R}^\infty$.

We denote by $\mathbb{R}^\infty$ the disjoint union $\bigsqcup_{n\geq 1}\mathbb{R}^n$. This is the space where inputs are taken from in numerical computations and, in fact, a BSS-machine M computes a (partial) function $\varphi^M : \mathbb{R}^\infty \to \mathbb{R}^\infty$.

In case M computes $\varphi^M : \mathbb{R}^\infty \to \{0,1\}$ we say that M *decides* the set $S := \{a \in \mathbb{R}^\infty \mid \varphi(a) = 1\}$. For $a \in \mathbb{R}^\infty$ we write $|a| = n$ when $a \in \mathbb{R}^n$ and we say that n is the *size* of a.

We denote by $\mathbb{R}^\infty$ the disjoint union $\bigsqcup_{n\geq 1}\mathbb{R}^n$. This is the space where inputs are taken from in numerical computations and, in fact, a BSS-machine M computes a (partial) function $\varphi^M : \mathbb{R}^\infty \to \mathbb{R}^\infty$.

In case M computes $\varphi^M : \mathbb{R}^\infty \to \{0,1\}$ we say that M *decides* the set $S := \{a \in \mathbb{R}^\infty \mid \varphi(a) = 1\}$. For $a \in \mathbb{R}^\infty$ we write $|a| = n$ when $a \in \mathbb{R}^n$ and we say that n is the *size* of a.

The class $\mathrm{P}_\mathbb{R}$ is then defined as the class of sets S for which there is a machine M deciding S and such that, for every $a \in \mathbb{R}^\infty$, the cost $\mathrm{cost}^M(a)$ of the computation of M with input a is polynomial in $|a|$. That is,

$$\mathrm{P}_\mathbb{R} := \{S \subseteq \mathbb{R}^\infty \mid \exists M\, \forall a\, (\varphi^M(a) = 1 \iff a \in S)\ \&\ \mathrm{cost}^M(a) = |a|^{\mathcal{O}(1)}\}.$$

We denote by $\mathbb{R}^\infty$ the disjoint union $\bigsqcup_{n\geq 1}\mathbb{R}^n$. This is the space where inputs are taken from in numerical computations and, in fact, a BSS-machine M computes a (partial) function $\varphi^M : \mathbb{R}^\infty \to \mathbb{R}^\infty$.

In case M computes $\varphi^M : \mathbb{R}^\infty \to \{0,1\}$ we say that M *decides* the set $S := \{a \in \mathbb{R}^\infty \mid \varphi(a) = 1\}$. For $a \in \mathbb{R}^\infty$ we write $|a| = n$ when $a \in \mathbb{R}^n$ and we say that n is the *size* of a.

The class $\mathrm{P}_\mathbb{R}$ is then defined as the class of sets S for which there is a machine M deciding S and such that, for every $a \in \mathbb{R}^\infty$, the cost $\mathrm{cost}^M(a)$ of the computation of M with input a is polynomial in $|a|$. That is,

$$\mathrm{P}_\mathbb{R} := \{S \subseteq \mathbb{R}^\infty \mid \exists M\,\forall a\,(\varphi^M(a) = 1 \iff a \in S) \;\&\; \mathrm{cost}^M(a) = |a|^{\mathcal{O}(1)}\}.$$

We can similarly define the class $\mathrm{NP}_\mathbb{R}$. A set S is in $\mathrm{NP}_\mathbb{R}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathrm{P}_\mathbb{R}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \exists y\,(a, y) \in B.$$

The point $y \in \mathbb{R}^\infty$ is said the be a *witness* for the membership of a to S.

An example of set in $\mathrm{NP}_{\mathbb{R}}$ is the set 4FEAS of polynomials of degree 4 in several variables which possess a real zero. In this case, the input a corresponds to a polynomial system f in n variables and the witness y to a point $\xi \in \mathbb{R}^n$. The computation performed by M amounts to evaluating $f(\xi)$ and returning 1 if and only if this evaluation yields zero. The size of f (in this case, the number of coefficients of f) is $\Theta(n^4)$ and the evaluation can be done with cost polynomial in this size.

An example of set in $\mathrm{NP}_{\mathbb{R}}$ is the set 4FEAS of polynomials of degree 4 in several variables which possess a real zero. In this case, the input a corresponds to a polynomial system f in n variables and the witness y to a point $\xi \in \mathbb{R}^n$. The computation performed by M amounts to evaluating $f(\xi)$ and returning 1 if and only if this evaluation yields zero. The size of f (in this case, the number of coefficients of f) is $\Theta(n^4)$ and the evaluation can be done with cost polynomial in this size.

More importantly, Blum, Shub, and Smale proved that 4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete pointing towards a fundamental difficulty in the problem of deciding whether a polynomial (or a polynomial system) has a real zero.

An example of set in $\mathrm{NP}_{\mathbb{R}}$ is the set 4FEAS of polynomials of degree 4 in several variables which possess a real zero. In this case, the input a corresponds to a polynomial system f in n variables and the witness y to a point $\xi \in \mathbb{R}^n$. The computation performed by M amounts to evaluating $f(\xi)$ and returning 1 if and only if this evaluation yields zero. The size of f (in this case, the number of coefficients of f) is $\Theta(n^4)$ and the evaluation can be done with cost polynomial in this size.

More importantly, Blum, Shub, and Smale proved that 4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete pointing towards a fundamental difficulty in the problem of deciding whether a polynomial (or a polynomial system) has a real zero.

Unlike the situation in the context of discrete computations, there was no avalanche of $\mathrm{NP}_{\mathbb{R}}$-complete problems as a sequel of the BSS paper. A simple reason is a smaller pool of problems (reasonable candidates will have to be problems dealing with polynomial systems and sets defined by them). A deeper reason, which explains why, nonetheless, problems in this pool had not been classified as complete in *any* complexity class over the reals is the fact that the catalog of complexity classes over the reals was not broad enough.

An immediate extension of the definition of $\mathrm{NP}_{\mathbb{R}}$ is the following.

Definition

Let $\mathcal{C}$ be a complexity class of decision problems. A set S is in $\exists\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \exists y\, (a, y) \in B.$$

A set S is in $\forall\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \forall y\, (a, y) \in B.$$

An immediate extension of the definition of $\mathrm{NP}_{\mathbb{R}}$ is the following.

Definition

Let $\mathcal{C}$ be a complexity class of decision problems. A set S is in $\exists\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \exists y \, (a, y) \in B.$$

A set S is in $\forall\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \forall y \, (a, y) \in B.$$

The class $\mathrm{NP}_{\mathbb{R}}$ is therefore $\exists\mathrm{P}_{\mathbb{R}}$ and we will also denote it simply by $\exists$. Also, the class $\mathrm{coNP}_{\mathbb{R}}$ is defined to be $\forall\mathrm{P}_{\mathbb{R}}$ and we will denote it by $\forall$.

An immediate extension of the definition of $\mathrm{NP}_{\mathbb{R}}$ is the following.

Definition

Let $\mathcal{C}$ be a complexity class of decision problems. A set S is in $\exists\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \exists y\,(a, y) \in B.$$

A set S is in $\forall\mathcal{C}$ if and only if there exists $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in S \iff \forall y\,(a, y) \in B.$$

The class $\mathrm{NP}_{\mathbb{R}}$ is therefore $\exists\mathrm{P}_{\mathbb{R}}$ and we will also denote it simply by $\exists$. Also, the class $\mathrm{coNP}_{\mathbb{R}}$ is defined to be $\forall\mathrm{P}_{\mathbb{R}}$ and we will denote it by $\forall$.

This definition allows to alternate the use of the quantifiers $\exists$ and $\forall$ to obtain classes such as $\exists\forall\exists$, which is usually denoted by $\Sigma^3_{\mathbb{R}}$. The union of all the complexity classes obtained this way, starting from $\mathrm{P}_{\mathbb{R}}$, is the *polynomial hierarchy* over $\mathbb{R}$.

The classes in the polynomial hierarchy over $\mathbb{R}$ are not sufficient to accommodate many of the problems naturally arising when considering polynomial systems and the sets they define. We next describe some of these problems.

The classes in the polynomial hierarchy over $\mathbb{R}$ are not sufficient to accommodate many of the problems naturally arising when considering polynomial systems and the sets they define. We next describe some of these problems.

An *algebraic circuit* over $\mathbb{R}$ is an acyclic directed graph where each node has indegree 0, 1 or 2. Nodes with indegree 0 are either labeled as *input nodes* or with elements of $\mathbb{R}$ (we shall call them *constant nodes*). Nodes with indegree 2 are labeled with the binary operators of $\mathbb{R}$, i.e. one of $\{+, \times, -, /\}$. They are called *arithmetic nodes*. Nodes with indegree 1 are either *sign nodes* or *output nodes*.

The classes in the polynomial hierarchy over $\mathbb{R}$ are not sufficient to accommodate many of the problems naturally arising when considering polynomial systems and the sets they define. We next describe some of these problems.

An *algebraic circuit* over $\mathbb{R}$ is an acyclic directed graph where each node has indegree 0, 1 or 2. Nodes with indegree 0 are either labeled as *input nodes* or with elements of $\mathbb{R}$ (we shall call them *constant nodes*). Nodes with indegree 2 are labeled with the binary operators of $\mathbb{R}$, i.e. one of $\{+, \times, -, /\}$. They are called *arithmetic nodes*. Nodes with indegree 1 are either *sign nodes* or *output nodes*.

To a circuit $\mathscr{C}$ with n input nodes and m output nodes one associates a function $f_{\mathscr{C}} : \mathbb{R}^n \to \mathbb{R}^m$. This function may not be total since divisions by zero may occur (in which case $f_{\mathscr{C}}$ is not defined on its input).

We say that $\mathscr{C}$ is a *decision* circuit if $f_{\mathscr{C}} : \mathbb{R}^n \to \{0, 1\}$. The set *decided* by the circuit is

$$S_{\mathscr{C}} = \{a \in \mathbb{R}^n \mid f_{\mathscr{C}}(a) = 1\}.$$

Subsets of $\mathbb{R}^n$ decidable by algebraic circuits are known as *semialgebraic sets*. They can be written as a Boolean combination of solution sets of polynomial inequalities $\{a \in \mathbb{R}^n \mid f(a) \geq 0\}$.

We say that $\mathscr{C}$ is a *decision* circuit if $f_{\mathscr{C}} : \mathbb{R}^n \to \{0,1\}$. The set *decided* by the circuit is

$$S_{\mathscr{C}} = \{a \in \mathbb{R}^n \mid f_{\mathscr{C}}(a) = 1\}.$$

Subsets of $\mathbb{R}^n$ decidable by algebraic circuits are known as *semialgebraic sets*. They can be written as a Boolean combination of solution sets of polynomial inequalities $\{a \in \mathbb{R}^n \mid f(a) \geq 0\}$.

Semialgebraic sets will be inputs to our problems. They will be given either by a Boolean combination of polynomial equalities and inequalities or by a decision circuit. If not otherwise specified, we mean the first variant. In this case, polynomials are encoded with the so called *dense encoding*, i.e., they are represented by the complete list of their coefficients (including zero coefficients, as in 4FEAS above).

Partial functions $f : \mathbb{R}^n \to \mathbb{R}^m$ computable by algebraic circuits are known as *piecewise rational*. These are the functions f for which there exists a semialgebraic partition $\mathbb{R}^n = S_0 \cup S_1 \cup \ldots \cup S_k$ and rational functions $g_i : S_i \to \mathbb{R}^m$, $i = 1, \ldots, k$ such that g_i is well-defined on S_i and $f_{|S_i} = g_i$. Note that f is undefined on S_0. We will also consider piecewise rational functions as inputs to some problems. They will be encoded by algebraic circuits.

We next define the following problems:

We next define the following problems:

$\mathrm{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

We next define the following problems:

$\mathrm{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

$\mathrm{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

We next define the following problems:

$\mathrm{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.
$\mathrm{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.
$\mathrm{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.

We next define the following problems:

$\mathrm{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.
$\mathrm{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.
$\mathrm{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.
$\mathrm{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.

We next define the following problems:

$\text{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.
$\text{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.
$\text{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.
$\text{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.
$\text{ERD}_{\mathbb{R}}$ (*Euclidean Relative Denseness*) Given semialgebraic sets S and V, decide whether S is included in $\overline{V}$.

We next define the following problems:

$\mathrm{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

$\mathrm{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

$\mathrm{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.

$\mathrm{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.

$\mathrm{ERD}_{\mathbb{R}}$ (*Euclidean Relative Denseness*) Given semialgebraic sets S and V, decide whether S is included in $\overline{V}$.

$\mathrm{LERD}_{\mathbb{R}}$ (*Linearly restricted Euclidean Relative Denseness*) Given a semialgebraic set $V \subseteq \mathbb{R}^n$ and points $a_0, a_1, \ldots, a_k \in \mathbb{R}^n$, decide whether $a_0 + \langle a_1, \ldots, a_k \rangle$ is included in $\overline{V}$.

We next define the following problems:

$\text{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

$\text{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

$\text{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.

$\text{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.

$\text{ERD}_{\mathbb{R}}$ (*Euclidean Relative Denseness*) Given semialgebraic sets S and V, decide whether S is included in $\overline{V}$.

$\text{LERD}_{\mathbb{R}}$ (*Linearly restricted Euclidean Relative Denseness*) Given a semialgebraic set $V \subseteq \mathbb{R}^n$ and points $a_0, a_1, \ldots, a_k \in \mathbb{R}^n$, decide whether $a_0 + \langle a_1, \ldots, a_k \rangle$ is included in $\overline{V}$.

$\text{ZADH}_{\mathbb{R}}$ (*Zariski Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Zariski closure $\overline{S}^Z$ of S.

We next define the following problems:

$\text{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

$\text{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

$\text{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.

$\text{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.

$\text{ERD}_{\mathbb{R}}$ (*Euclidean Relative Denseness*) Given semialgebraic sets S and V, decide whether S is included in $\overline{V}$.

$\text{LERD}_{\mathbb{R}}$ (*Linearly restricted Euclidean Relative Denseness*) Given a semialgebraic set $V \subseteq \mathbb{R}^n$ and points $a_0, a_1, \dots, a_k \in \mathbb{R}^n$, decide whether $a_0 + \langle a_1, \dots, a_k \rangle$ is included in $\overline{V}$.

$\text{ZADH}_{\mathbb{R}}$ (*Zariski Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Zariski closure $\overline{S}^Z$ of S.

$\text{ZDENSE}_{\mathbb{R}}$ (*Zariski Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}}^Z = \mathbb{R}^n$.

We next define the following problems:

$\text{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a polynomial $f \in \mathbb{R}[X_1, \dots, X_n]$, decide whether there exists $x \in \mathbb{R}^n$ such that $f(x) = 0$.

$\text{DIM}_{\mathbb{R}}(d)$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

$\text{EADH}_{\mathbb{R}}$ (*Euclidean Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Euclidean closure $\overline{S}$ of S.

$\text{EDENSE}_{\mathbb{R}}$ (*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}} = \mathbb{R}^n$.

$\text{ERD}_{\mathbb{R}}$ (*Euclidean Relative Denseness*) Given semialgebraic sets S and V, decide whether S is included in $\overline{V}$.

$\text{LERD}_{\mathbb{R}}$ (*Linearly restricted Euclidean Relative Denseness*) Given a semialgebraic set $V \subseteq \mathbb{R}^n$ and points $a_0, a_1, \dots, a_k \in \mathbb{R}^n$, decide whether $a_0 + \langle a_1, \dots, a_k \rangle$ is included in $\overline{V}$.

$\text{ZADH}_{\mathbb{R}}$ (*Zariski Adherence*) Given a semialgebraic set S and a point x, decide whether x belongs to the Zariski closure $\overline{S}^Z$ of S.

$\text{ZDENSE}_{\mathbb{R}}$ (*Zariski Denseness*) Given a decision circuit $\mathscr{C}$ with n input nodes, decide whether $\overline{S_{\mathscr{C}}}^Z = \mathbb{R}^n$.

$\text{UNBOUNDED}_{\mathbb{R}}$ (*Unboundedness*) Given a semialgebraic set S, is it unbounded?

$\textsc{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?

$\textsc{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\textsc{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\text{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?

$\textsc{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\textsc{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\textsc{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\textsc{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?
$\textsc{BasicCompact}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?

$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.

$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.

$\text{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?

$\text{BasicCompact}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?

$\text{SOCS}_{\mathbb{R}}(k)$ (*Smallest Order Coefficient Sign, k variables*) Given a division-free straight-line program Γ in k input variables $X_1, \ldots, X_k$, decide whether the smallest-order coefficient (w.r.t. the ordering $X_1 \succ X_2 \succ \ldots \succ X_k$) of f_Γ (the polynomial in X computed by Γ) is positive.

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\text{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?
$\text{BasicCompact}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?
$\text{SOCS}_{\mathbb{R}}(k)$ (*Smallest Order Coefficient Sign, k variables*) Given a division-free straight-line program Γ in k input variables $X_1, \ldots, X_k$, decide whether the smallest-order coefficient (w.r.t. the ordering $X_1 \succ X_2 \succ \ldots \succ X_k$) of f_Γ (the polynomial in X computed by Γ) is positive.
$\text{LocSupp}_{\mathbb{R}}$ (*Local Support*) Given a circuit $\mathscr{C}$ with n input nodes and a linear equation $\ell(x) = 0$, decide whether there exists $x_0 \in \mathbb{R}^n$ and $\delta > 0$ such that $S_{\mathscr{C}} \cap \{\ell < 0\} \cap B(x_0, \delta) = \emptyset$ and $\dim(\overline{S_{\mathscr{C}}} \cap \{\ell = 0\} \cap B(x_0, \delta)) = n - 1$.

$\text{LOCDIM}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{ISOLATED}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{EXISTISO}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\text{BASICCLOSED}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?
$\text{BASICCOMPACT}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?
$\text{SOCS}_{\mathbb{R}}(k)$ (*Smallest Order Coefficient Sign, k variables*) Given a division-free straight-line program Γ in k input variables $X_1, \ldots, X_k$, decide whether the smallest-order coefficient (w.r.t. the ordering $X_1 \succ X_2 \succ \ldots \succ X_k$) of f_Γ (the polynomial in X computed by Γ) is positive.
$\text{LOCSUPP}_{\mathbb{R}}$ (*Local Support*) Given a circuit $\mathscr{C}$ with n input nodes and a linear equation $\ell(x) = 0$, decide whether there exists $x_0 \in \mathbb{R}^n$ and $\delta > 0$ such that $S_{\mathscr{C}} \cap \{\ell < 0\} \cap B(x_0, \delta) = \emptyset$ and $\dim(\overline{S_{\mathscr{C}}} \cap \{\ell = 0\} \cap B(x_0, \delta)) = n - 1$.
$\text{TOTAL}_{\mathbb{R}}$ (*Totalness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is total.

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\text{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?
$\text{BasicCompact}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?
$\text{SOCS}_{\mathbb{R}}(k)$ (*Smallest Order Coefficient Sign, k variables*) Given a division-free straight-line program Γ in k input variables $X_1, \dots, X_k$, decide whether the smallest-order coefficient (w.r.t. the ordering $X_1 \succ X_2 \succ \dots \succ X_k$) of f_Γ (the polynomial in X computed by Γ) is positive.
$\text{LocSupp}_{\mathbb{R}}$ (*Local Support*) Given a circuit $\mathscr{C}$ with n input nodes and a linear equation $\ell(x) = 0$, decide whether there exists $x_0 \in \mathbb{R}^n$ and $\delta > 0$ such that $S_{\mathscr{C}} \cap \{\ell < 0\} \cap B(x_0, \delta) = \emptyset$ and $\dim(\overline{S_{\mathscr{C}}} \cap \{\ell = 0\} \cap B(x_0, \delta)) = n - 1$.
$\text{Total}_{\mathbb{R}}$ (*Totalness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is total.
$\text{Inj}_{\mathbb{R}}$ (*Injectiveness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is injective.

$\text{LocDim}_{\mathbb{R}}$ (*Local Dimension*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, a point $x \in S$, and $d \in \mathbb{N}$, is $\dim_x S \geq d$?
$\text{Isolated}_{\mathbb{R}}$ (*Isolated*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, decide whether x is an isolated point of S.
$\text{ExistIso}_{\mathbb{R}}$ (*Existence of isolated points*) Given a semialgebraic set $S \subseteq \mathbb{R}^n$, decide whether there exist a point x isolated in S.
$\text{BasicClosed}_{\mathbb{R}}$ (*Closedness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it closed?
$\text{BasicCompact}_{\mathbb{R}}$ (*Compactness for basic semialgebraic sets*) Given a basic semialgebraic set S, is it compact?
$\text{SOCS}_{\mathbb{R}}(k)$ (*Smallest Order Coefficient Sign, k variables*) Given a division-free straight-line program Γ in k input variables $X_1, \ldots, X_k$, decide whether the smallest-order coefficient (w.r.t. the ordering $X_1 \succ X_2 \succ \ldots \succ X_k$) of f_Γ (the polynomial in X computed by Γ) is positive.
$\text{LocSupp}_{\mathbb{R}}$ (*Local Support*) Given a circuit $\mathscr{C}$ with n input nodes and a linear equation $\ell(x) = 0$, decide whether there exists $x_0 \in \mathbb{R}^n$ and $\delta > 0$ such that $S_{\mathscr{C}} \cap \{\ell < 0\} \cap B(x_0, \delta) = \emptyset$ and $\dim(\overline{S_{\mathscr{C}}} \cap \{\ell = 0\} \cap B(x_0, \delta)) = n - 1$.
$\text{Total}_{\mathbb{R}}$ (*Totalness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is total.
$\text{Inj}_{\mathbb{R}}$ (*Injectiveness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is injective.
$\text{Surj}_{\mathbb{R}}$ (*Surjectiveness*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is surjective.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.

ImageZDense$_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
ImageEDense$_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{ImageEDense}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{DomainZDense}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{ImageEDense}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{DomainZDense}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{DomainEDense}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{ImageEDense}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{DomainZDense}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{DomainEDense}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{Cont}_{\mathbb{R}}$ (*Continuity*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.

$\text{IMAGEZDENSE}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\text{IMAGEEDENSE}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\text{DOMAINZDENSE}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.
$\text{DOMAINEDENSE}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.
$\text{CONT}_{\mathbb{R}}$ (*Continuity*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.
$\text{CONT}_{\mathbb{R}}^{\text{DF}}$ (*Continuity for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{ImageEDense}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{DomainZDense}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{DomainEDense}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{Cont}_{\mathbb{R}}$ (*Continuity*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.
$\textsc{Cont}_{\mathbb{R}}^{\mathrm{DF}}$ (*Continuity for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.
$\textsc{ContPoint}_{\mathbb{R}}^{\mathrm{DF}}$ (*Continuity at a Point for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$ with n input nodes and $x \in \mathbb{R}^n$, decide whether $f_{\mathscr{C}}$ is continuous at x.

$\textsc{ImageZDense}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{ImageEDense}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{DomainZDense}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.
$\textsc{DomainEDense}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.
$\textsc{Cont}_{\mathbb{R}}$ (*Continuity*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.
$\textsc{Cont}_{\mathbb{R}}^{\mathrm{DF}}$ (*Continuity for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.
$\textsc{ContPoint}_{\mathbb{R}}^{\mathrm{DF}}$ (*Continuity at a Point for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$ with n input nodes and $x \in \mathbb{R}^n$, decide whether $f_{\mathscr{C}}$ is continuous at x.
$\textsc{Lipschitz}_{\mathbb{R}}(k)$ (*Lipschitz-k*) Given a circuit $\mathscr{C}$, and $k > 0$, decide whether $f_{\mathscr{C}}$ is Lipschitz-k, i.e., whether for all $x, y \in \mathbb{R}^n$, $\|f(x) - f(y)\| \leq k\|x - y\|$.

$\text{IMAGEZDENSE}_{\mathbb{R}}$ (*Image Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Zariski dense.

$\text{IMAGEEDENSE}_{\mathbb{R}}$ (*Image Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the image of $f_{\mathscr{C}}$ is Euclidean dense.

$\text{DOMAINZDENSE}_{\mathbb{R}}$ (*Domain Zariski Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Zariski dense.

$\text{DOMAINEDENSE}_{\mathbb{R}}$ (*Domain Euclidean Dense*) Given a circuit $\mathscr{C}$, decide whether the domain of $f_{\mathscr{C}}$ is Euclidean dense.

$\text{CONT}_{\mathbb{R}}$ (*Continuity*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.

$\text{CONT}_{\mathbb{R}}^{\text{DF}}$ (*Continuity for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is continuous.

$\text{CONTPOINT}_{\mathbb{R}}^{\text{DF}}$ (*Continuity at a Point for Division-Free Circuits*) Given a division-free circuit $\mathscr{C}$ with n input nodes and $x \in \mathbb{R}^n$, decide whether $f_{\mathscr{C}}$ is continuous at x.

$\text{LIPSCHITZ}_{\mathbb{R}}(k)$ (*Lipschitz-k*) Given a circuit $\mathscr{C}$, and $k > 0$, decide whether $f_{\mathscr{C}}$ is Lipschitz-k, i.e., whether for all $x, y \in \mathbb{R}^n$, $\|f(x) - f(y)\| \leq k\|x - y\|$.

$\text{LIPSCHITZ}_{\mathbb{R}}$ (*Lipschitz*) Given a circuit $\mathscr{C}$, decide whether $f_{\mathscr{C}}$ is Lipschitz, i.e., whether there exists $k > 0$ such that $f_{\mathscr{C}}$ is Lipschitz-k.

To classify the complexity of these problems some new complexity classes, with no counterparts in the discrete context, are necessary.

To classify the complexity of these problems some new complexity classes, with no counterparts in the discrete context, are necessary.

Definition

Let $\mathcal{C}$ be a complexity class of decision problems. A set A belongs to $\mathsf{H}\mathcal{C}$ if there exists $B \subseteq \mathbb{R} \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in A \iff \exists \mu > 0 \, \forall \varepsilon \in (0, \mu) \, (\varepsilon, a) \in B.$$

A set A belongs to $\forall^* \mathcal{C}$ if there exist a polynomial p and a set $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in A \iff \dim\{z \in \mathbb{R}^{p(|a|)} \mid (z, a) \notin B\} < p(|a|).$$

If $\mathcal{C}$ is a complexity class we denote by $\mathcal{C}^{\mathsf{c}}$ the class of its complements, i.e., the class of all sets A such that $A^{\mathsf{c}} \in \mathcal{C}$. We define $\exists^* \mathcal{C} := (\forall^* \mathcal{C}^{\mathsf{c}})^{\mathsf{c}}$.

We note that A belongs to $\exists^*\mathcal{C}$ if and only if there exist a polynomial p and a set $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in A \iff \dim\{z \in \mathbb{R}^{p(|a|)} \mid (z,a) \in B\} = p(|a|).$$

We note that A belongs to $\exists^*\mathcal{C}$ if and only if there exist a polynomial p and a set $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

$$a \in A \iff \dim\{z \in \mathbb{R}^{p(|a|)} \mid (z, a) \in B\} = p(|a|).$$

The quantifier H was introduced in [BC09]. It captures the notion of "for all sufficiently small numbers." The quantifiers $\forall^*$ and $\exists^*$ capture the notions of "for almost all points" and "for sufficiently many points" in a specific sense. They were first introduced by P. Koiran.

We note that A belongs to $\exists^*\mathcal{C}$ if and only if there exist a polynomial p and a set $B \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$, $B \in \mathcal{C}$, such that, for all $a \in \mathbb{R}^\infty$,

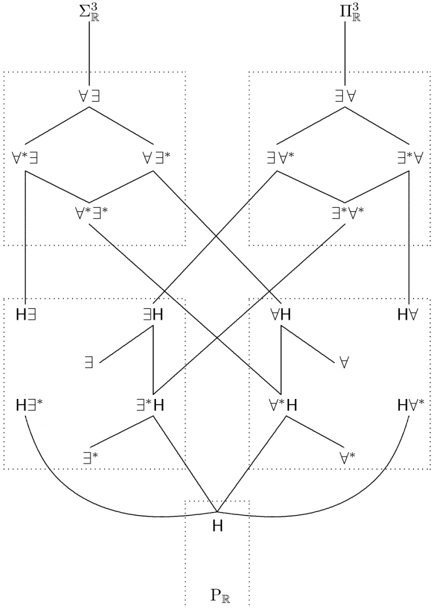
$$a \in A \iff \dim\{z \in \mathbb{R}^{p(|a|)} \mid (z,a) \in B\} = p(|a|).$$

The quantifier H was introduced in [BC09]. It captures the notion of "for all sufficiently small numbers." The quantifiers $\forall^*$ and $\exists^*$ capture the notions of "for almost all points" and "for sufficiently many points" in a specific sense. They were first introduced by P. Koiran.

Using these operators we may define many new complexity classes. Notations such as $\exists^*\forall$, H$\forall$, or $\exists^*$H denote some of the newly created complexity classes in an obvious manner. To avoid a cumbersome notation, we also write H instead of $\mathrm{HP}_{\mathbb{R}}$.
It is beyond the goal of this talk to describe in detail all of the structural properties of the classes defined above. We just mention that H is closed by complements and that, for any class $\mathcal{C}$ as above, we have the inclusion

$$\exists^*\mathcal{C} \subseteq \exists\mathcal{C}.$$

The following diagram gives a landscape of complexity classes in the lower levels of the polynomial hierarchy over $\mathbb{R}$. All upward lines mean inclusion. Note that not all possible classes below $\Sigma^3_{\mathbb{R}}$ or $\Pi^3_{\mathbb{R}}$ are in the diagram. We restricted attention to those standing out (e.g., because of having natural complete problems).
Boxes enclosing groups of complexity classes do not have a very formal meaning. They are rather meant to convey the informal idea that some classes are "close enough" to be clustered together.


$\Sigma^3_{\mathbb{R}}$
$\Pi^3_{\mathbb{R}}$
$\forall\exists$
$\exists\forall$
$\forall^*\exists$
$\forall\exists^*$
$\exists\forall^*$
$\exists^*\forall$
$\forall^*\exists^*$
$\exists^*\forall^*$
$\mathsf{H}\exists$
$\exists\mathsf{H}$
$\forall\mathsf{H}$
$\mathsf{H}\forall$
$\exists$
$\forall$
$\mathsf{H}\exists^*$
$\exists^*\mathsf{H}$
$\forall^*\mathsf{H}$
$\mathsf{H}\forall^*$
$\exists^*$
$\forall^*$
H
$\mathrm{P}_{\mathbb{R}}$

We finally summarize the complexity sorting of the list of problems given above in the following table. The meaning of each row should be clear.

We finally summarize the complexity sorting of the list of problems given above in the following table. The meaning of each row should be clear.

Problem	Complete in	Lower bound	Upper bound
$\text{FEAS}_{\mathbb{R}}$	$\exists$		
$\text{DIM}_{\mathbb{R}}(d)$	$\exists$		
$\text{SOCS}_{\mathbb{R}}(k)$	H^k		
$\text{ZDENSE}_{\mathbb{R}}$	$\exists^*$		
$\text{DOMAINZDENSE}_{\mathbb{R}}$	$\exists^*$		
$\text{EDENSE}_{\mathbb{R}}$	$\forall^*$		
$\text{DOMAINEDENSE}_{\mathbb{R}}$	$\forall^*$		
$\text{IMAGEZDENSE}_{\mathbb{R}}$	$\exists$		
$\text{TOTAL}_{\mathbb{R}}$	$\forall$		
$\text{INJ}_{\mathbb{R}}$	$\forall$		
$\text{LIPSCHITZ}_{\mathbb{R}}(k)$	$\forall$		

Problem	**Complete in**	**Lower bound**	**Upper bound**
$\textsc{ZAdh}_{\mathbb{R}}$		$\exists$	?
$\textsc{EAdh}_{\mathbb{R}}$	$\mathsf{H}\exists$		
$\textsc{Unbounded}_{\mathbb{R}}$	$\mathsf{H}\exists$		
$\textsc{LocDim}_{\mathbb{R}}$	$\mathsf{H}\exists$		
$\textsc{Isolated}_{\mathbb{R}}$	$\mathsf{H}\forall$		
$\textsc{Cont}_{\mathbb{R}}$		$\forall$	$\mathsf{H}^3\forall$
$\textsc{Cont}_{\mathbb{R}}^{\mathrm{DF}}$		$\forall$	$\mathsf{H}^2\forall$
$\textsc{ContPoint}_{\mathbb{R}}^{\mathrm{DF}}$	$\mathsf{H}\forall$		
$\textsc{Lipschitz}_{\mathbb{R}}$		$\forall$	$\mathsf{H}\forall$
$\textsc{LocSupp}_{\mathbb{R}}$	$\exists^*\mathsf{H}$		
$\textsc{ExistIso}_{\mathbb{R}}$		$\mathsf{H}\forall$	$\exists\forall$
$\textsc{BasicClosed}_{\mathbb{R}}$	$\mathsf{H}\forall$		
$\textsc{BasicCompact}_{\mathbb{R}}$	$\mathsf{H}\forall$		
$\textsc{LERD}_{\mathbb{R}}$	$\forall^*\exists$		
$\textsc{ImageEDense}_{\mathbb{R}}$	$\forall^*\exists$		
$\textsc{ERD}_{\mathbb{R}}$		$\forall^*\exists$	$\forall\exists$
$\textsc{Surj}_{\mathbb{R}}$	$\forall\exists$		